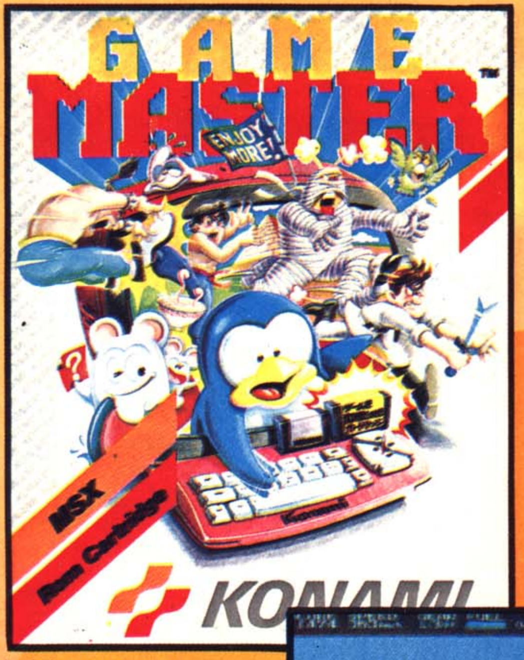

INPUT
Publicación práctica para usuarios de MSX
Revista mensual 1987
Año 2-Número 17 Precio 375 Ptas.
MSX
MAPAS Y CARGADORES
DE LAST MISSION,
ARMY MOVES Y
HOWARD EL PATO
SPRITES EN
CODIGO MAQUINA
CONSTRUYE TU
TURBO-JOYSTICK







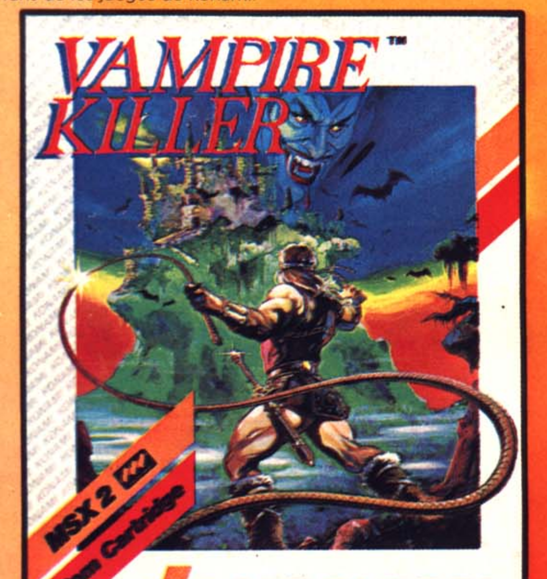


GAMESMASTER

Gamesmaster es la única respuesta para esos juegos difíciles.
Posibilidades de ralentizar el movimiento e incluso congelar el juego, modificar la velocidad y etapas del juego.
Volcados de pantalla por impresora, pasa a cinta o a disco las máximas puntuaciones. Selecciona el número de jugadores y calcula las máximas puntuaciones.
En realidad, algo esencial para los fans de los juegos de Konami.

PENGUIN ADVENTURE

Guía a nuestro héroe Penguin por cuevas, mares y glaciares para devolver la normalidad a la Princesa Penguin y su reino.
Pelea con los Tiranosaurios y con montones de extraños enemigos utilizando los poderes comprados al Comerciante Esquimal.
Apuesta los peces en una máquina tragaperras para aumentar la puntuación y bucea por escenarios submarinos en un intento de restaurar el Paraíso Penguin.
Un juego lleno de acción con los gráficos que acostumbras esperar de Konami.



ULTIMAS NOVEDADES KONAMI SERMA





VAMPIRE KILLER

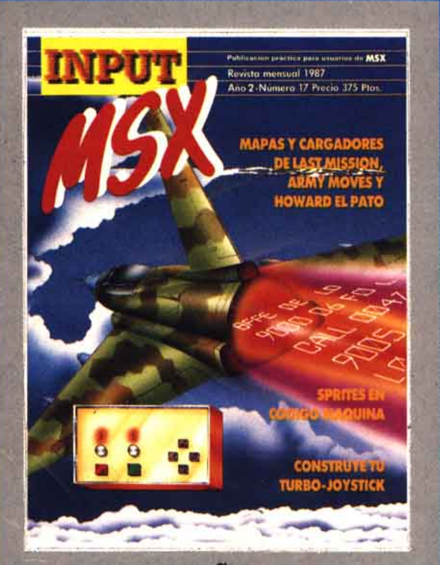
Enfréntate a Drácula en esta espectacular aventura de Konami. Atraviesa el camino del Diablo, consigue las armas y poderes especiales y quizá puedas atravesar el castillo satánico y luchar con el Maestro de la Oscuridad.
En cada esquina habrá adversarios que te helarán la sangre; necesitarás lograr los medios de superarlos. Una decisión equivocada te hará fracasar...
Este cartucho ROM con un Mega lleno de acción, con los brillantes gráficos a que Konami nos tiene acostumbrados, es una aventura del principio al final.



Vampire Killer - 6.800 pts. (solo MSX 2)
Penguin Adventure - 6.150 pts.
Game Master - 6.150 pts.

ENVIA ESTE CUPON A KONAMI SHOP. FRANCISCO NAVACERRADA, 19. 28028 MADRID.
TITULO:
NOMBRE Y APELLIDOS:
DIRECCION:
COD. POSTAL:
POBLACION:
PROVINCIA:
FORMA DE PAGO: CONTRARREMBOLSO ☐
POR TALON BANCARIO ☐

KONAMI ES DISTRIBUIDO EXCLUSIVAMENTE EN ESPAÑA POR SERMA: CARDENAL BELLUGA, 21. 28028 MADRID. TLFOS: 256 21 01 - 12 22.



AÑO 2 NUMERO 17

DIRECTOR: Manuel Pérez
REDACTOR JEFE: Antonio Pliego
REDACCION: Jaime Mardones
REALIZACION GRAFICA: Didac Tudela
COORDINADOR DE SOFT: Xavier Ferrer
MAPAS Y POKES: José Vila
COLABORADORES: Ernesto del Valle, Ramón Rabasó, Daniel C. Lepekhine, Juanma Ponce, Julio García, Javier de la Fuente, Equipo Molisoft, Irene Alcaraz, Angels Alvarez.
FOTOGRAFIA: Joan Boada

INPUT MSX es una publicación de PLANETA-DE AGOSTINI, S.A.

GERENTE DIVISION DE REVISTAS: Sebastián Martínez
DIRECTOR DE ARTE: Luis F. Balaguer

PUBLICIDAD: INTERMEDIA, S.A. Gral. Moscardó, 5 - 3.ª A 28020 MADRID. Teléf. (91) 442 70 44
BUFETE DE AGENTES DE PUBLICIDAD, S.A.: Plza. Alfonso X el Sabio, 7, 1.º, 4.ª Barcelona. Teléf. (93) 347 59 00

FOTOMECANICA: UNGRAF, S.A.
IMPRESION: Sirven Grafic c/. Gran Vía, 754-756. 08013 Barcelona
Depósito legal: B. 38.115-1986
SUSCRIPCIONES: EDISA López de Hoyos, 141. 28002 Madrid Teléf. (91) 415 97 12
REDACCION: Aribau, 185, 1.º 08021 Barcelona
DISTRIBUIDORA: R.B.A. PROMOTORA DE EDICIONES, S.A. Calle B, n.º 11. Sector B, Zona Franca 08004 Barcelona

El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.
INPUT MSX es independiente y no está vinculada a los distribuidores del estándar.
INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a través de las secciones adecuadas en estas páginas.



SUMARIO

EDITORIAL 4

TRUCOS 5

ACTUALIDAD 6

PROGRAMACION
APRENDE A DIBUJAR EN 3-D (y II) 10
DETECCION DE OBJETOS EN PANTALLA 19

HARDWARE
CONSTRUYE TU PROPIO JOYSTICK 22

CODIGO MAQUINA
MOVIENDO SPRITES EN C/M 25

SOFTWARE
TODO SOBRE HOWARD EL PATO 28
MAPA DE YIE AR KUNG-FU 2 41
MAPA DE ARMY MOVES 2 44
MAPA DE LAST MISSION 47

REVISTA DE SOFTWARE 52

EL ZOCO DE INPUT 65

PROGRAMACION DE JUEGOS (COLECCIONABLE)
JUGANDO A LA GUERRA (y II) 31



MAS NOVEDADES

Con este número de INPUT MSX termina la publicación del coleccionable Programación de Juegos *que ha ocupado las páginas centrales de nuestra revista desde su primer número.*

Tal y como ya sabrán quienes practiquen la sana costumbre de leer los editoriales de INPUT MSX, hace un par de números anunciamos la futura publicación de un cuadernillo mensual, no coleccionable, exclusivamente dedicado a Código Máquina. Lamentablemente, para que esto sea posible se impone que el coleccionable desaparezca. Sabemos que muchos lectores lo lamentarán y considerarán tal decisión injustificada.

A pesar de todo, nuestra opción ha sido la que muchos de vosotros veníais demandando desde hace tiempo: más atención hacia un tema que en nuestro estándar siempre ha quedado relegado a un lugar secundario, tanto por parte de los fabricantes como de sus publicaciones. Como veréis el mes que viene, su contenido vale la pena.

En otro orden de cosas queremos resaltar la muy favorable acogida que habéis dispensado a los nuevos concursos y a la fe de erratas. Erais bastantes los que os habíais quedado atascados por no tenerla y muchos también los que habéis colaborado, con vuestros descubrimientos, para que su publicación fuera posible.

Para terminar, tan sólo nos queda recordaros que la Redacción espera anhelante el correo que llega cada día, lanzándose vorazmente a su lectura a fin de conocer vuestras opiniones, críticas, hallazgos y propuestas. Por lo tanto, ¡ánimo y a escribir!

TRUCOS DEL USUARIO DE MSX

En esta oportunidad los trucos se refieren a programación de joysticks en código máquina y a atractivas representaciones y efectos de tipo caligráfico.

Entre los seleccionados de cada mes sortearemos tres interesantes videojuegos, siempre de la más rabiosa actualidad.

Si tenéis algún «truquillo» olvidado en el fondo del ordenador, mandadlo a:

INPUT MSX
C/ Aribau, 185, 1.º
08021-Barcelona

PROGRAMACION DE JOYSTICKS EN C/M

Los joysticks se controlan mediante los registros 15 y 14 del PSG. El registro 15 selecciona el port del que se va a leer; una vez hecho esto, podemos leer en el registro 14 el estado de dicho joystick.

Registro 15:

Bit 0: puesto a 1, activa el disparo 1 del joystick 1.
Bit 1: puesto a 1, activa el disparo 2 del joystick 1.
Bits 2 y 3: igual que los anteriores, pero para el joystick 2.
Bits 4, 5 y 6: seleccionan el joystick (1 o 2) que se va a leer, de la siguiente manera:

	B6	B5	B4
JOYSTICK 1	Ø	x	Ø
JOYSTICK 2	1	Ø	x

(Los bits marcados con una «x» no importan)
Bit 7: no se usa para la programación de los joysticks.

Registro 14:

Bit 0: se pone a cero si el joystick correspondiente (seleccionado en el registro 15) se mueve hacia arriba.
Bit 1: igual que el anterior, pero para el movimiento hacia abajo.
Bit 2: igual, para el movimiento hacia la izquierda.
Bit 3: igual, para la derecha.

Las diagonales ponen a cero los dos bits correspondientes (arriba-derecha, arriba-izquierda, etc.).
Bit 4: se pone a cero si de dispara el botón 1.
Bit 5: igual, pero para el botón 2.
Bit 6 y 7: no se usan.

Los que no quieran complicaciones pueden emplear las rutinas de la ROM. La de lectura del joystick comienza en la dirección &HD5, se carga el acumulador con el n.º de joystick (0-2), y la rutina devuelve en el acumulador la posición del joystick (0-8).

La de disparo comienza en &HD8, se carga el acumulador con el n.º de botón de disparo (0-4) y la rutina devuelve 255 si se está pulsando el disparador y 0 en caso contrario.

MIGUEL ANGEL FERNANDEZ GUTIERREZ
(Avilés)

CARACTERES Y COLORES

```
1Ø 'Representación de todos
15 'los caracteres del MSX
2Ø FORI=1TO255:OUT&H98,I:NEXT
25 GOTO25
```

```
1Ø 'Representación de todos
15 'los colores del MSX
2Ø DEFINTI,P
25 SCREEN3
3Ø FORI=ØTO1535
35 VPOKEI,P/8:P=P+2
4Ø IFP=256THENP=Ø
45 NEXT
5Ø GOTO5Ø
```

G. J. ACUÑA FUENTES
(Cartagena)

FANTASIA CALIGRAFICA

Trabajando con el ordenador en SCREEN 1 y con algunos, VPOKEs se obtienen interesantes efectos con los caracteres MSX. Éstos cambian su forma y se convierten en letras fantasía.

```
1Ø REM LETRAS FANTASIA 1
2Ø REM POR RICARD MARTI
3Ø REM PARA INPUT MSX.
4Ø SCREEN 1:CLS:KEY OFF:COLOR 15,4,4
5Ø LOCATE 6,1Ø:PRINT"ESPERA UN MOMENTO"
6Ø FOR K=256 TO 2Ø39
7Ø VPOKE K,VPEEK(K) XOR VPEEK (K)/4
8Ø NEXT K
9Ø END
```

Trabajando también en SCREEN 1 y con una variante del primer listado, obtenemos este interesante programa:

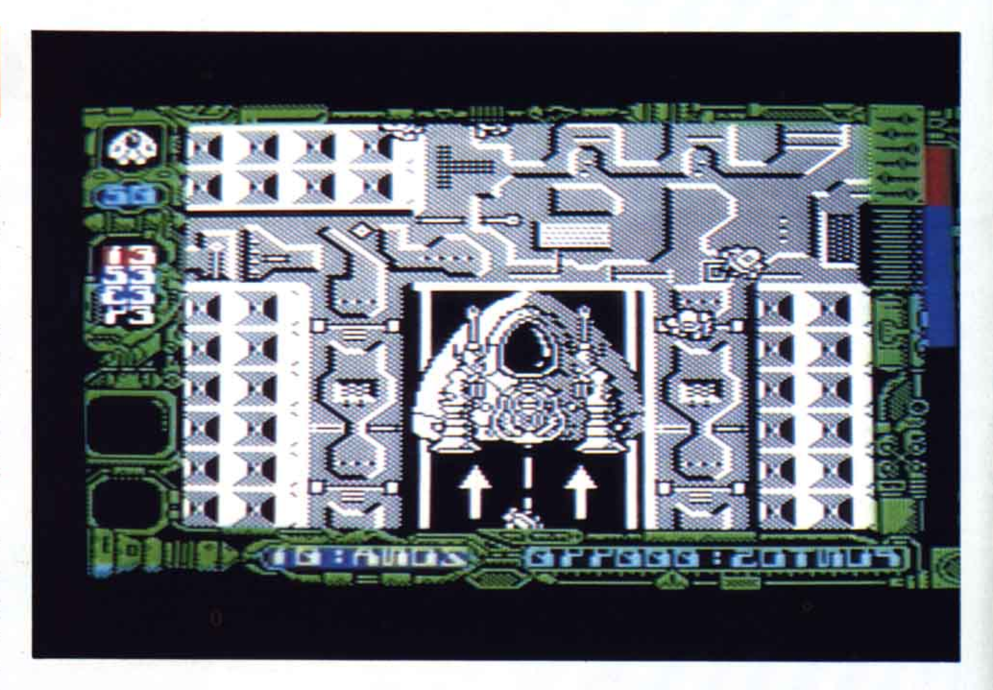
```
1Ø REM LETRAS FANTASIA 2
2Ø REM POR RICARD MARTI
3Ø REM PARA INPUT MSX
4Ø SCREEN 1:CLS:KEY OFF:COLOR 15,4,4
5Ø LOCATE 6,10:PRINT"ESPERA UN MOMENTO"
6Ø FOR K=1ØØ TO 1ØØØ
7Ø VPOKE K, VPEEK(K) OR VPEEK(K)/2+2
8Ø NEXT K
9Ø END
```

RICARD MARTI RIERA
SALLENT (Barcelona)

CINCO NOVEDADES DE DINAMIC EN MSX

Entre finales del mes de septiembre y las próximas Navidades, DINAMIC publicará nada menos que cinco títulos en versión MSX, la mayoría de los cuales ya fueron presentados con anterioridad en otros sistemas. Se trata de ARQUIMEDES XXI, COBRAS ARC, y DUSTIN, a los que se unirán más tarde (hacia diciembre) NAVY MOVES (la continuación del famoso ARMY MOVES) y FREDDY HARDEST.

Por otra parte, se mantiene la expectación sobre el lanzamiento de DON QUIJOTE, la nueva aventura gráfico-conversacional con que DINAMIC pretende ocupar durante mucho tiempo el "número uno" en todas las listas. Según nos confirmó recientemente el portavoz del grupo, se están recibiendo gran cantidad de peticiones para que se publique la versión MSX de este programa, y, aunque todavía no hay nada definitivo, nos ha asegurado que "están en ello". No obstante, existen grandes dificultades para adaptar el programa al estándar MSX, y por ello es presumible que, si llega a hacerse tal versión, no la veamos hasta bien entrado el próximo año.



OTRA VEZ TOPO

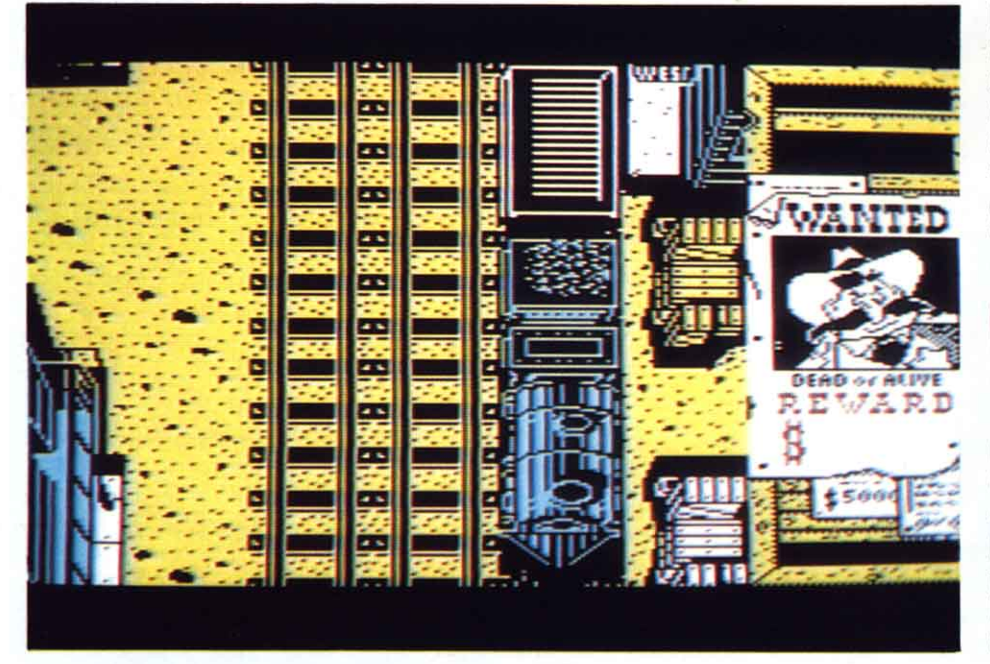
TOPO SOFT vuelve a la carga con dos nuevos programas, STARDUST y BANG, que serán presentados próximamente de cara a la campaña navideña.

Por el momento, sólo os podemos adelantar que se trata de dos arcades de desarrollo vertical, en los que se incorporan interesantes innovaciones técnicas, como el *scroll* a tres niveles superpuestos, con el que se logra un impresionante efecto tridimensional, o el avance del mapeado con *scroll* simultáneo de gráficos y atributos.

El tema será de ambientación futurista-espacial en un caso, y del más clásico estilo Far West en el otro, aunque ambos programas estarán presididos por un denominador común: una calidad gráfica digna del más rotundo "SOBRESALIENTE".

Estad atentos a su publicación porque, sin duda, van a arrollar.

Aunque en el momento de redactar estas líneas aún no se han terminado las versiones definitivas (los *masters*), a continuación publicamos algunas fotografías de las pantallas más espectaculares, algunas de las cuales, a pesar de ser casi perfectas tal y como ya están, deberán sufrir algunos retoques antes de que nuestros perfeccionistas amigos de TOPO las den por determinadas.



OPERA SOFT A LA CONQUISTA DE EUROPA

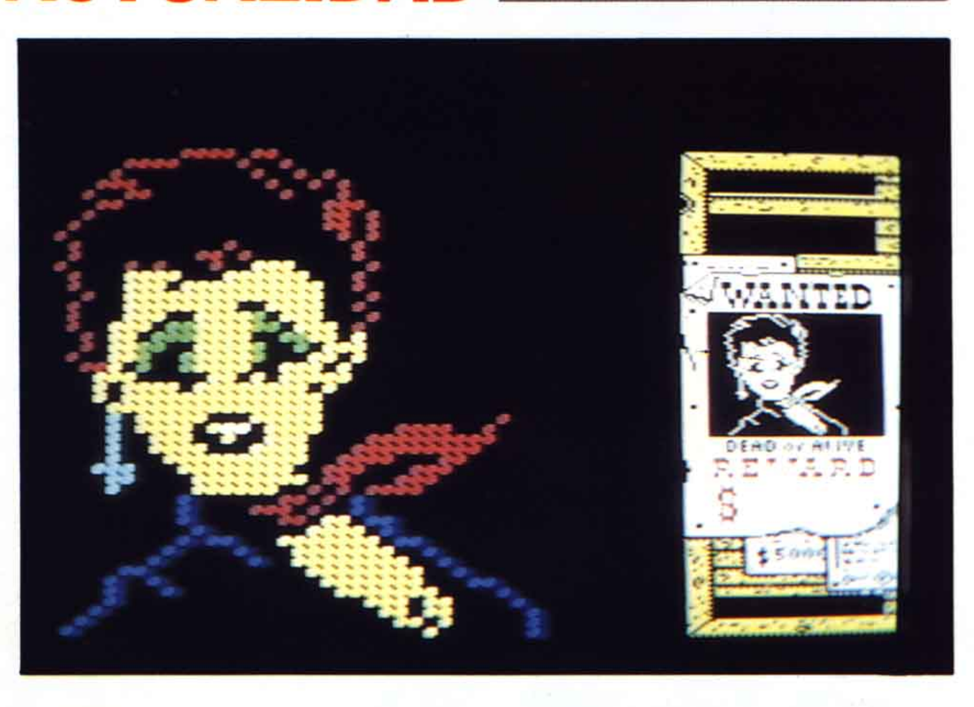
Con el título de "LIVINGSTONE.. I PRESUME?", la firma británica ALLIGATA SOFTWARE acaba de publicar en el Reino Unido la versión para MSX-1 (disponible en disco y cassette) de "LIVINGSTONE, SUPONGO", la sensacional vídeo-aventura con que nuestros compatriotas de OPE-

RA SOFT se dieron a conocer en España hace ahora algo menos de un año, y a la que dedicamos puntualmente un detallado comentario con mapa fotográfico y cargador de vidas infinitas (ver INPUT MSX n.º 11).

La versión "inglesa" del programa sólo se diferencia de la original en el idioma, que, lógicamente, ha sido traducido, y en la carátula de presentación, que ha sido sustituida por un nuevo dibujo donde, con un estilo de rasgos más caricaturescos, se da una expresiva imagen del contenido del juego (si hemos de ser sinceros, nos gustaba más la portada de Laugi).

Pocas semanas después del inicio de la campaña de promoción preparada por ALLIGATA (anuncios en revistas, pósters, etc.), las cifras de ventas del programa van camino de convertirle en un nuevo "número uno", como lo fuera en su día ROCKY de DINAMIC con la diferencia de que esta vez (¡por fin!) se hace mención expresa de la procedencia del juego: el nombre de OPERA SOFT, y la imagen de software hecho en España, ya resuenan en los oídos de los usuarios británicos.

¡Ojalá sirva de precedente!



MSX SIGUE SUBIENDO

Aunque muchos usuarios no lo saben, al estándar MSX le corresponde uno de los índices de ventas más bajos del mercado británico, tanto en hard como en soft, mientras que en nuestro país ocurre precisamente lo contrario. Quienes sí lo saben muy bien, son los fabricantes que, desde sus sedes en Japón o el Reino Unido, han contemplado asombrados el fenómeno: lo que hasta hace poco no era más que un mercado sólo "potencial" –el español– ahora se ha convertido en una importante fuente de beneficios, incluso a través de productos que teóricamente estarían fuera del alcance de nuestros "tan poco europeos bolsillos", como los cartuchos MSX-2, que rondan las cinco mil pesetas.

Por este motivo, se han repetido a lo largo del año frecuentes visitas y acuerdos sobre operaciones comerciales con distribuidores españoles, evidenciando el gran interés que despiertan las enormes posibilidades de nuestro mercado.

Como ejemplo, diremos que en MASTERTRONIC ya no se ruborizan al reconocer que su producción para MSX se hace pensando en el mercado español.

Luther de Gale, representante de KONAMI en Gran Bretaña, nos comentaba que las causas fundamentales de éxito del sistema MSX en nuestro país son dos:

– Por un lado, la incorporación de este estándar al mercado español antes de que otros sistemas llegaran a consolidarse plenamente (como ocurrió en el Reino Unido con Spectrum, Amstrad y Commodore);

– Por otro, una eficaz promoción y un precio competitivo.

En otras palabras, podríamos decir que MSX no tuvo inicialmente más competencia que la de Spectrum, ya que tanto Commodore como Amstrad presentaron el inconveniente de ser más caros y de estar insuficientemente promocionados, aunque en la actualidad esta tendencia se ha corregido.

Afortunadamente, los usuarios españoles de MSX somos cada vez más, y esto va propiciando que poco a poco aumente también el volumen de títulos publicados para este estándar en relación a otros sistemas, superando así ese lastre de la escasez de software que hasta hace poco pesaba sobre nosotros.

WALTER MILLER STRIKES BACK!

Después del éxito obtenido en la comercialización del compatibilizador MULTIMILLER y de la ampliación de memoria de 64 K, MEMORY MILLER, se confirma la pronta aparición en el mercado de una memoria RAM de 256 K, útil para aquellos programas megaenormes.

MULTIMILLER será dentro de pocos meses un cartucho clásico para aquellos ordenadores "incompatibles", tanto de la primera como de la segunda generación. Los SONY y PHILIPS de la segunda ya no tendrán problemas con los programas reseteadores. MULTIMILLER llegará a ser la horma de nuestro zapato-binario-electrónico.

MEMORY MILLER, como hemos subrayado anteriormente, es una ampliación de memoria de 64 K a un precio increíble. Esperamos que se vendan como churros y a sus creadores los felicitamos para que sigan en la misma línea de más por menos. Es decir: bueno, bonito y barato.

METAL GEAR

KONAMI prepara en España el lanzamiento de METAL GEAR, un trepidante programa de ambientación bélica en el que se retoma el planteamiento clásico del héroe que lucha solo contra las hordas enemigas.

El objetivo del juego consiste en superar un gran número de fases sucesivas en creciente nivel de dificultad, infiltrándose en las líneas enemigas y destruyendo sus defensas. Para lograrlo, es preciso utilizar las armas adecuadas en los momentos precisos, siempre y cuando se haya encontrado previamente el lugar donde se esconden los más preciados pertrechos. Por orden de importancia, éstos son:

- PISTOLA;
- PISTOLA AMETRALLADORA;
- FUSIL DE PRECISION;
- BAZOOKA;
- BOMBA;
- MISIL y
- MINA

Además del arsenal de rigor, existen una serie de objetos que, usados correctamente, pueden facilitar la misión de forma considerable, como el chaleco antibalas (¡imprescindible!), la brújula o la linterna.

Resumiendo, podemos definir a METAL GEAR como una interesante mezcla de dos conocidos ingredientes: grandes sobredosis de acción con planteamiento de arcade, junto a un desarrollo argumental propio de la más enrevesada videoaventura.

El programa será puesto a la venta en España coincidiendo con la campaña navideña, aunque por el momento no se puede aventurar una fecha concreta de publicación. En su momento, INPUT MSX os ofrecerá un detallado comentario sobre el mismo.

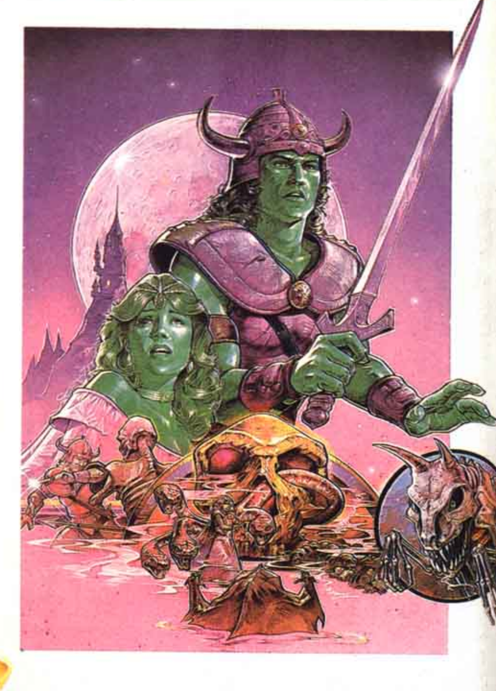
THE MAZE OF GALIOUS

Entre las últimas novedades presentadas por KONAMI, destaca la esperada segunda parte del arcade KNIGHTMARE, publicada con el nombre de MAZE OF GALIOUS en fecha muy reciente, y aún no distribuida oficialmente en España.



En esta nueva aventura, el héroe deberá superar un complejo laberinto de pantallas estructuradas en plataformas de escaleras, hasta llegar al corazón del Castillo Mágico y derrotar al malvado GALIOUS, sumo sacerdote de una secta diabólica. El grado de dificultad no es tan elevado como suele ser habitual en los juegos que llevan la firma de KONAMI, y el número total de pantallas no alcanza la cifra de otros programas en formato cartucho, pero, a pesar de todo, os podemos asegurar que MAZE OF GALIOUS posee todos los elementos necesarios para crear una persistente adicción, sin llegar a desesperar a los menos hábiles.

Como recordaréis, en la primera parte de la historia el guerrero Poppolon debía rescatar a la bella Afrodita de las garras de los temibles Hudnos; lo que no sabía entonces nuestro esforzado héroe es que todo había sido una trampa preparada por Galious para desviar su atención y poder raptar impunemente a su hijo Pampas. Ahora, Poppolon y Afrodita se dirigen al tenebroso Castillo Griego de Galious, con la intención de liberar a su hijo a cualquier precio...



A diferencia de la primera parte, en esta ocasión no todo consiste en combatir y avanzar: además, habrá que recoger armas, llaves y otros objetos, algunos de los cuales serán imprescindibles para culminar con éxito la aventura. Este detalle, unido a un diseño gráfico más espectacular y variado, y a un planteamiento de mayor originalidad, contribuye a hacer de MAZE OF GALIOUS una «segunda parte» que supera con creces a la primera.

En cuanto a la calidad de los gráficos, debemos señalar un aspecto importante: a medida que se avanza en el juego, los escenarios van ganando progresivamente en espectacularidad y profusión, incorporando sugestivas «sorpresas» y eliminando rápidamente una impresión inicial menos positiva.

En suma, un excelente programa en formato cartucho compatible con MSX-1, que viene a confirmar la calidad del sello que lo presenta. De nuevo KONAMI refuerza su imagen de prestigio.

Software

JAMES BOND 007

IN

THE LIVING DAYLIGHTS

THE COMPUTER GAME



ALBERT R. BROCCOLI
Presents
TIMOTHY DALTON
as IAN FLEMING'S
JAMES BOND 007

Amstrad CPC/PCW
CBM 64/Amiga
Spectrum 48/128/Plus 2/3
BBC B & Master
Atari 8 Bit
MSX

DOMARK

THE LIVING DAYLIGHTS

Starring MARYAM d'ABO JOE DON BAKER ART MALIK and JEROEN KRABBÉ
Production Designer PETER LAMONT Music by JOHN BARRY Associate Producers TOM PEVSNER and BARBARA BROCCOLI
Produced by ALBERT R. BROCCOLI and MICHAEL G. WILSON Directed by JOHN GLEN Screenplay by RICHARD MAIBAUM and MICHAEL G. WILSON

TECHNICOLOR® PANAVISION® DOLBY STEREO

ORIGINAL SOUNDTRACK ALBUM AVAILABLE ON WARNER BROS. RECORDS, CASSETTES AND COMPACT DISCS

TITLE SONG PERFORMED BY a-ha

UA United Artists

APRENDE A DIBUJAR EN 3-D (y II)

El programa de dibujo de modelos de alambre que hemos visto en los dos artículos anteriores te permite dibujar un cubo en tres dimensiones y cambiar su tamaño. Pero te presenta siempre el mismo punto de vista; la cara frontal del cubo es la que aparece siempre más próxima a ti y, aunque puedes ver a través del sólido de alambre, no tienes posibilidad de verlo desde arriba, desde un lado o desde donde quieras mirarlo. En este artículo encontrarás unas rutinas adicionales que te permitirán especificar la posición de tu ojo de manera que puedas contemplar el cubo desde cualquier dirección.

Se trata de una herramienta muy útil, especialmente cuando empieces a dibujar objetos más complicados, ya que al contemplarlos según diferentes direcciones puedes descubrir rasgos ocultos o semiescondidos. Además, si especificas una sucesión de coordenadas diferentes para la posición de tu ojo, puedes obtener la impresión de estar *sobrevolando* el objeto o caminando alrededor del mismo. Sin embargo, la secuencia de dibujos que es posible obtener en un ordenador doméstico es muy lenta y el efecto no es el mismo que se obtiene con la velocidad de los sólidos de alambre comerciales en que el observador parece que se precipita sobre el coche, planeta o lo que sea, a gran velocidad. No obstante, los principios que intervienen son básicamente los mismos.

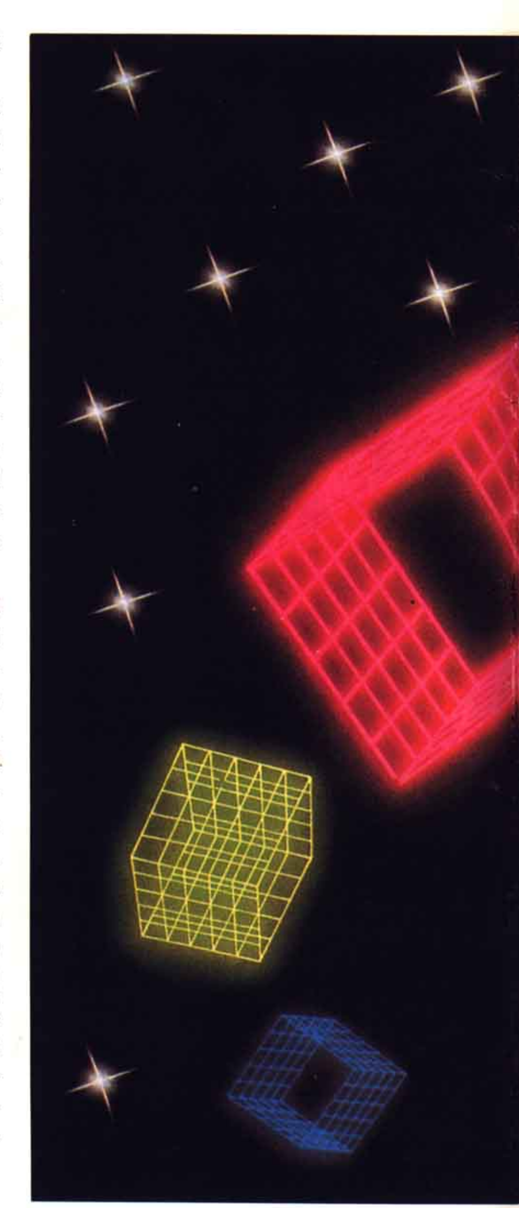
EL EFECTO DEL PUNTO DE VISTA

Ya vimos en el artículo anterior cómo es posible dibujar un objeto *tridimensional* en una pantalla de dos dimensiones utilizando una convención visual que puede ser interpretada por el ojo y el cerebro observador. Los programas anteriores hacían esto utilizando una proyección *isométrica* en la que las líneas inclinadas se supone que retroceden o avanzan a partir de la pantalla.

La forma más común de convención visual es el dibujo en perspectiva, en el que las líneas que se alejan del observador convergen en un punto distante, llamado punto de fuga, acortándose en escorzo como si *desaparecieran*. En una verdadera perspectiva de tres puntos hay realmente tres direcciones de fuga y tres puntos de fuga, una para cada uno de los tres ejes. La posición de los tres puntos está determinada por la relación entre la posición del observador y la posición del objeto que se desea representar. (Recuerda que en la convención visual estos puntos son imaginarios.)

Con esto ya disponemos de una clave sobre la forma de representar un objeto, tal como un cubo en un espacio de tres dimensiones, tal como se vería desde diferentes puntos de vista. Para ello será necesario definir unas nuevas transformaciones que produzcan el efecto de convergencia hacia un punto de fuga y que acorten el objeto.

Lo primero que tienes que saber es la relación que hay entre la posición del observador y la del objeto. Para determinarla tienes que relacionar el punto de vista con los ejes X, Y y Z que has definido en tu pantalla. Eso es precisamente lo que hacen los programas que vamos a ver, ocupándose después de las transformaciones necesarias.



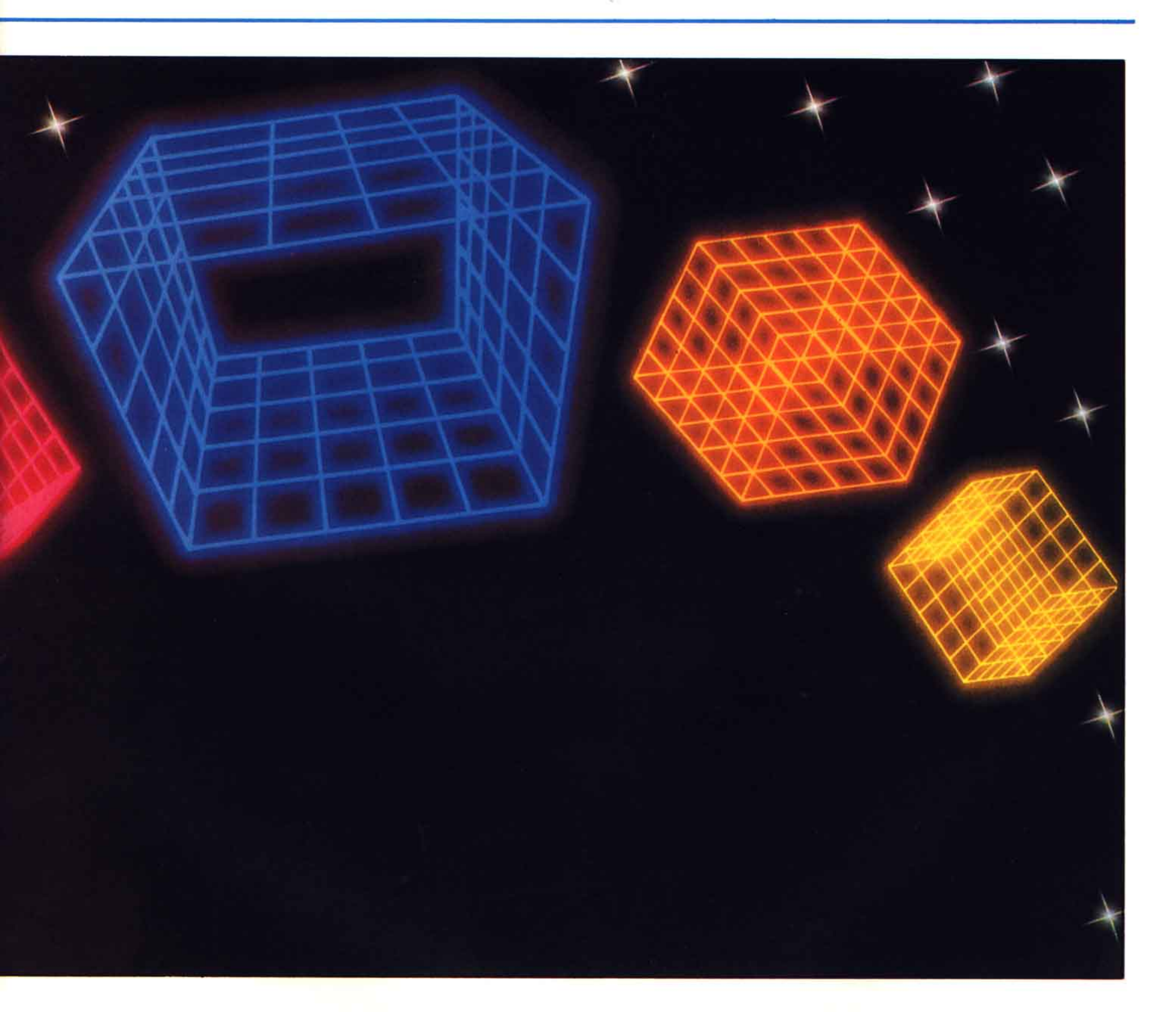
ANTES DE EMPEZAR

Para poder ejecutar las nuevas secciones del programa, necesitas la *rutina de rejilla* que vimos en el artículo anterior. Si guardaste una copia de dicha rutina, tendrás ahora que cargarla con una instrucción LOAD. Sería una buena idea que cargaras también la *rutina de dibujo del círculo* que también explicamos. En realidad ahora no la vas a necesitar, pero conviene tener todas las rutinas juntas para el programa de dibujo del globo que veremos más adelante. En consecuencia las líneas que necesitas son desde la 5000 a la 6160; puedes borrar todas las demás líneas.

Igual que antes, tendrás que teclear varias secciones de programa antes de poder pasar a ejecutar nada, ya que

¡Impresionante! Ésta es la única manera de calificar la potencia transformadora de este programa, el segundo de la serie, en el que se incorporan la perspectiva, el punto de vista y la generación de figuras alámbricas.

- EL EFECTO DEL PUNTO DE VISTA
- ESTABLECIENDO LAS VARIABLES INICIALES
- PROGRAMACION MODULAR



primeramente tienes que definir la posición del ojo y las transformaciones para las rotaciones y la perspectiva.

DEFINIENDO TU PUNTO DE VISTA

La primera sección del programa calcula las variables necesarias para determinar la posición de tu ojo en el espacio tridimensional:

```
8ØØØ XV=X;YV=Y:ZV=Z
8Ø1Ø WV=YV*YV+ZV*ZV
8Ø2Ø PV=SQR(XV*XV+WV)
8Ø3Ø IFPV=ØTHENRETURN
8Ø35 WV=SQR(WV)
8Ø4Ø XU=XV/PV
8Ø5Ø YU=YV/PV
8Ø6Ø ZU=ZV/PV
8Ø7Ø WU=WV/PV
8Ø8Ø REM ** ORIENTACION **
8Ø9Ø A=XV*YV:B=ZV:
     GOSUB845Ø:G=H
81ØØ A=YV:B=XV:
     GOSUB845Ø:G=G+H
811Ø SG=SIN(G)
812Ø CG=COS(G)
814Ø R1=WU*CG
```

```
8150 R2=-WU*SG
8160 R3=-XU
8170 R4=-YU
8180 R5=-ZU
8190 R6=XV*XU+YV*YU+ZV*ZU
8200 IFWU=0THEN GOTO8340
8210 XT=(XV*WU)-(YV*YU+ZV*ZU)*(XU/WU)
8220 YT=(YV*ZU-ZV*YU)/WU
8230 R7=(ZU*SG-XU*YU*CG)/WU
8240 R8=((-YU*SG)-XU*ZU*CG)/WU
8250 R9=CG*XT+SG*YT
8260 S1=(ZU*CG+XU*YU*SG)/WU
8270 S2=(-YU*CG+XU*ZU*SG)/WU
8280 S3=(-SG*XT)+CG*YT
8330 RETURN
8340 REM
8350 R7=-1
8360 R8=0
8370 R9=0
8380 S1=0
8390 S2=1
8400 S3=0
8410 RETURN
8450 IFB<>0THEN LETH=ATN(A/B):RETURN
8460 H=PI/2:RETURN
```

Para entender lo que va sucediendo, recuerda que el eje Z avanza hacia ti saliendo desde el centro de la pantalla, el eje Y apunta hacia arriba de la pantalla y el eje X apunta hacia la derecha de la pantalla.

El ojo se supone situado en (XV, YV, ZV), donde la V significa punto de Vista. La variable WV da la distancia en la dirección resultante de combinar Y y Z. Se supone que el objeto que estás contemplando está situado en el origen de coordenadas (0,0,0), origen que por simplicidad suponemos coincidente con el centro de la pantalla. La línea 8030 fuerza la salida de la rutina por la vía rápida en el caso en que el punto de vista, es decir la posición del ojo, sea el origen, ya que es difícil que mires a tu propio ojo, como no sea con un espejo. Las variables XU, YU y ZU son las distancias en las direcciones de los tres ejes X, Y, Z de una línea de longitud unidad dibujada entre el origen y la posición del ojo. WU es la distancia resultante en la dirección combinada de X y Z.

Cuando contemplas un objeto, puedes obtener diferentes vistas del mismo, moviéndote a su alrededor de izquierda a derecha. Para ello se mide un ángulo a partir del eje X, tal como se hace en las líneas 8090 y 8100. Hemos incluido una comprobación (líneas 8450 y 8460) para evitar la división por cero, que interrumpiría el programa y te enviaría un mensaje de error.

Con esto se obtiene una vista normal cuando se mira directamente a lo largo del eje X, produciéndose un giro gradual a medida que te vas moviendo alrededor, con lo que resultan vistas más interesantes. Las líneas 8140 a 8280 definen las variables que fijan la orientación del ojo en el espacio. La posición del ojo se fija de manera que su eje Z permanece sobre la línea que une el ojo con el punto de origen de la pantalla.

El resto de la rutina define las variables para el caso especial en que la posición del ojo está realmente sobre el eje X.

LAS TRANSFORMACIONES

La siguiente sección transforma las coordenadas X, Y, Z del cubo en coordenadas finales sobre la pantalla. Estas transformaciones tienen en cuenta la posición del ojo y el efecto de perspectiva:

```
8500 X1=T1*X+T4*Y+T7
8510 Y1=T2*X+T5*Y+T8
8520 Z1=T3*X+T6*Y+T9
8540 LETX2=R1*X1+R7*Y1+R8*Z1+R9
8550 LETY2=R2*X1+S1*Y1+S2*Z1+S3
8560 LETZ2=R3*X1+R4*Y1+R5*Z1+R6
8575 IFZ2<ZNTHEN RETURN
```





```
858Ø X3=D*X2/Z2
859Ø Y3=(-D*Y2)/Z2
86ØØ RETURN
```

Esta rutina utiliza el álgebra matricial, algo complicada, para transformar las coordenadas (X,Y), aunque básicamente se trata de los pasos de transformación descritos en el primer artículo de esta serie. Las líneas 8500 a 8520 transforman el plano (X,Y) de dibujo en dos dimensiones en el espacio (X1, Y1, Z1) de tres dimensiones. Las líneas 8540 a 8560 transforman las coordenadas (X1, Y1, Z1) del espacio tridimensional posicionándolas con arreglo a la situación y dirección del ojo (X2, Y2, Z2). La línea 8575 comprueba si la nueva posición que hay que dibujar está demasiado cerca de la posición del ojo, es decir, si Z2 se encuentra comprendido entre 0 y ZN, o si está detrás del ojo (Z2 negativo). En ambos casos la posición de dibujo se ignora, ya que sería imposible visualizar el objeto. Si la posición está más alejada que ZN, contada a partir de la posición del ojo, al final de la rutina se calculan las coordenadas de la posición sobre la pantalla (X3, Y3). El parámetro D/Z2 que figura en estas líneas incorpora la perspectiva a la imagen, al proyectar el objeto sobre una pantalla plana a una distancia D del ojo. Si asignas a D un valor pequeño, resultará el efecto de mover la pantalla cerca del ojo, con lo que el efecto de perspectiva es más acusado. Cuando D es grande, la pantalla está lejos y el efecto de perspectiva es pequeño, la imagen aparece virtualmente sin distorsión, incluso aun cuando se contempla oblicuamente en una dirección cualquiera.

ASIGNACION DE LAS VARIABLES INICIALES

El siguiente paso es escribir de nuevo las *rutinas de Dibujo* e *Inicialización* para poder hacer uso de las nuevas *rutinas de Transformación*:

```
9ØØØ CLS
9Ø2Ø XM=256:YM=192
9Ø3Ø XD=XM/2:YD=YM/2
9Ø4Ø ZN=1
9Ø42 INPUT"INTRODUCE LA
     DISTANCIA AL PLANO DE
     PROYECCION";D
9Ø45 IFD<=ØTHEN
     D=1ØØØ*ZN
9Ø5Ø T1=1:T2=Ø:T3=Ø
9Ø6Ø T4=Ø:T5=1:T6=Ø
9Ø7Ø T7=Ø:T8=Ø:T9=1
9Ø9Ø CLS:SCREEN2
91ØØ RETURN
95ØØ X=XS:Y=YS:GO
     SUB85ØØ:IFZ2<
     ZNTHENGOTO952Ø
95Ø5 IFX3<-127ORY3<
     -95ORX3>128ORY3>
     96THENGOTO955Ø
951Ø PSET (127+X3,95+Y3)
952Ø X=XE:Y=YE:GO
     SUB85ØØ:IFZ2<
     ZNTHENGOTO955Ø
9525 IFX3<-127ORY3<
     -95ORX3>128ORY3>
     96THENGOTO955Ø
953Ø LINE -(127+X3-XP,
     95+Y3-YP)
955Ø RETURN
```

La línea 9020 define las dimensiones máximas de la pantalla en las direcciones X e Y, mientras que la línea 9030 define el punto medio. La línea 9040 fija la posición más próxima al ojo, que se permite a los puntos de dibujo. La variable D da la distancia real desde la posición del ojo hasta el plano de proyección, es decir, hasta la pantalla, con lo que determina la perspectiva. Los valores de D se asignan durante la fase de ejecución del programa, con lo cual puedes modificar el grado de perspectiva. Si no se introduce ningún valor, al pulsar ENTER RETURN la línea 9045 establece un valor por defecto de 1000. Las líneas 9050 a 9070 se ocupan de pasar los valores a las constantes de transformación, para especificar en el espacio de tres dimensiones el plano en que hay que dibujar la imagen.

Esta sección del programa continúa con la versión revisada de la rutina de dibujo. En primer lugar se definen las constantes de la transformación (línea 9500) y a continuación se hace una comprobación (líneas 9505 y 9525) para determinar si hay que dibujar un nuevo punto en la pantalla (líneas 9510 y 9530). Esta comprobación se hace necesaria para evitar el mensaje de error que resultaría si intentas pintar la pantalla.

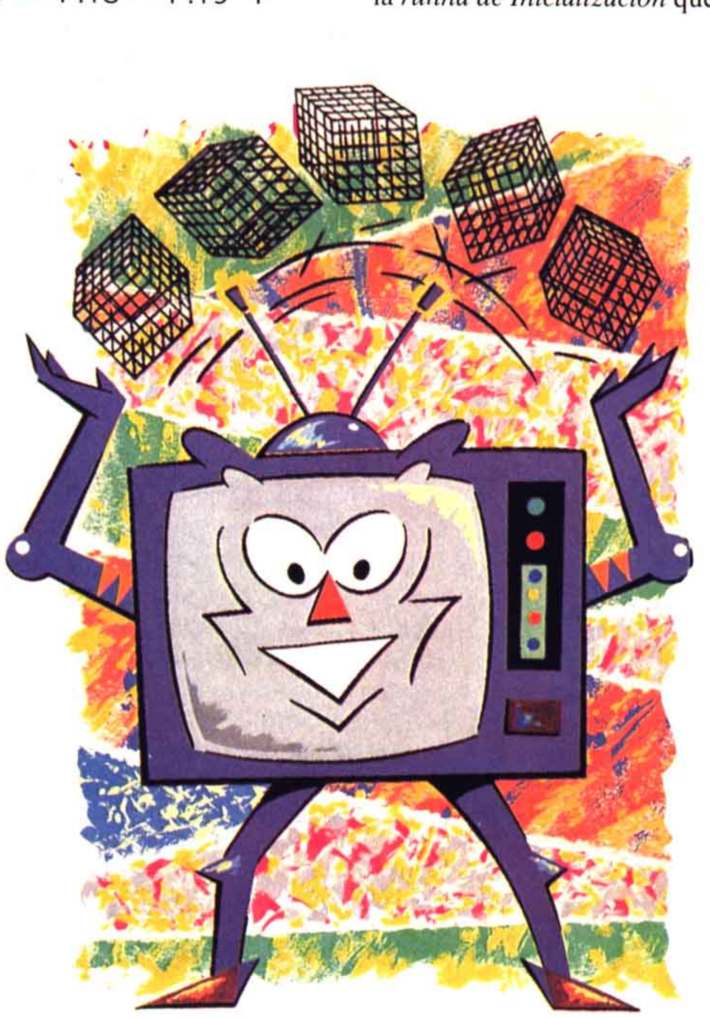
LLAMANDO A LA RUTINA

Ahora necesitas la rutina para dibujar la retícula. Si no tecleaste el programa del artículo anterior, hazlo ahora, junto con este programa.

```
1ØØ PI=3.1415927
11Ø GOSUB9ØØØ
12Ø L=2Ø:N=3
125 GOSUB5Ø5:GOTO14Ø
```

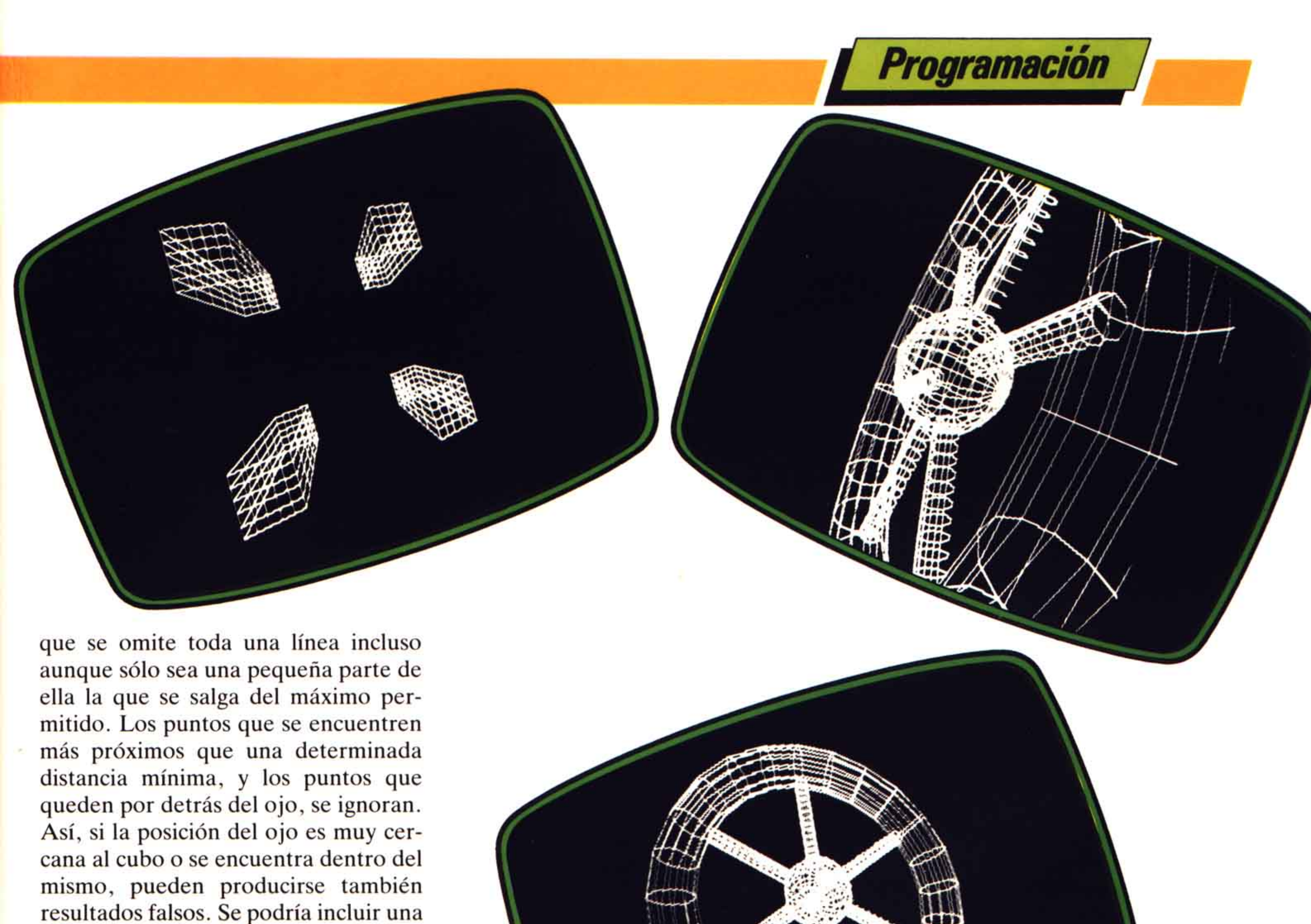
```
130 GOSUB500
140 IFX=0ANDY=0
    ANDZ=0THENGOTO170
150 GOSUB1000
160 GOTO130
170 CLS
180 END
500 BEEP:A$=INKEY$:IFA$=
    ""THENGOTO500
505 REM
510 INPUT"INTRODUCE LA
    POSICION DEL OJO (X,Y,Z)";
    X,Y,Z
515 SCREEN2
520 GOSUB8000
530 RETURN
1000 P=L/2
1010 T1=1:T2=0:T3=0
1020 T4=0:T5=1:T6=0
1030 T7=-P:T8=-P:T9=-P
1040 REM ** ABAJO **
1041 GOSUB1200
1050 T7=-P:T8=-P:T9=P
1060 REM ** ARRIBA **
1061 GOSUB1200
1070 T4=0:T5=0:T6=-1
1080 REM ** IZQUIERDA **
1081 GOSUB1200
1090 T7=-P:T8=P:T9=P
1100 REM ** DERECHA **
1101 GOSUB1200
1110 T1=0:T2=-1:T3=0
1120 REM ** ATRAS **
1121 GOSUB1200
1130 T7=P:T8=P:T9=P
1140 REM ** ADELANTE **
1141 GOSUB1200
1170 RETURN
1200 XA=0:YA=0:LW=L
1210 LH=L:NX=N:NY=N
1220 GOSUB5000
1230 RETURN
```

Puedes ejecutar ahora el programa. Si funciona correctamente, entrará a la *rutina de Inicialización* que empieza en la línea 110. Esta rutina define algunas variables y presenta un mensaje de pantalla pidiéndote que introduzcas un valor para D, la distancia al plano de proyección. La perspectiva incorporada en el programa es tal que cuanto mayor es esta distancia, más lejos aparece el objeto con lo que resultará menos efecto de perspectiva. Para empezar puedes introducir un valor de 1000. El programa retorna a la línea 120, que especifica la longitud L de cada lado del cubo, y el número N de cuadrados de la rejilla en cada cara. Las líneas 130 a 160 leen las coordenadas de la posición del ojo (línea 510), dibujando a continuación el cubo a partir de esa posición. En cuanto esté dibujada la primera vista, puedes introducir un nuevo conjunto de coordenadas, para ver el cubo desde una nueva dirección. El programa termina cuando se introducen los valores 0, 0 y 0. La rutina que abarca las líneas 500 a 530 hace realmente una llamada a la *rutina de Posición* (línea 520) para establecer las constantes de la transformación. La *rutina del Cubo* (líneas 1000 a 1170) es la que posiciona a continuación y dibuja cada una de las seis caras. La *rutina Cara* situada entre las líneas 1200 y 1220 dibuja cada cara como una cuadrícula, con ayuda de la *rutina Grid*.



Ensaya diferentes valores de la posición del ojo para estudiar el efecto del punto de vista. Para empezar, puedes utilizar un valor de 1000 para D y 200, 0 y 0 para X, Y, Z. El cubo aparece con el mismo tamaño pero la perspectiva es mucho más pronunciada. Para introducir nuevos valores de X, Y y Z, pulsa cualquier tecla y aparecerá el mensaje de pantalla. Para modificar el valor de D, tienes que pulsar BREAK y a continuación ejecutar el programa de nuevo. Te aparecerá un mensaje pidiéndote el valor de D y un segundo mensaje pidiéndote los valores de X, Y y Z.

Si en algún momento hubiera que dibujar puntos fuera de la zona de pantalla, te aparecerá un mensaje de error. En consecuencia el programa ignora cualquier línea que se salga fuera de la zona disponible. Esto puede llevar a resultados extraños, ya



que se omite toda una línea incluso aunque sólo sea una pequeña parte de ella la que se salga del máximo permitido. Los puntos que se encuentren más próximos que una determinada distancia mínima, y los puntos que queden por detrás del ojo, se ignoran. Así, si la posición del ojo es muy cercana al cubo o se encuentra dentro del mismo, pueden producirse también resultados falsos. Se podría incluir una comprobación de estos puntos, haciendo un recorte aproximado de las líneas, pero esto requeriría una considerable cantidad de programa y de cálculo adicional.

Guarda ahora en cinta o en disco una copia del listado completo, ya que a continuación verás la forma de usar estas mismas rutinas para dibujar formas duplicadas y para generar algunos gráficos circulares bastante espectaculares.

MODELOS DE ALAMBRE Y CURVAS

Hasta ahora hemos estado viendo la forma de conseguir figuras básicas a partir de retículas rectangulares y círculos concéntricos. También hemos visto cómo se pueden utilizar diferentes transformaciones para combinar una serie de rejillas que componen la imagen de un cubo en perspectiva. En este artículo introducimos en el programa que hemos desarrollado hasta ahora unas variaciones mínimas que nos permitirán ampliar la gama de posibles figuras de alambre que se pueden dibujar.

Para empezar, es muy sencillo conseguir imágenes múltiples. Si has trabajado con el programa que empieza en la línea 8000, habrás observado que se obtienen efectos de perspectiva bastante espectaculares cuando D (la distancia al observador) adquiere valores pequeños, por ejemplo entre 100 y 400, mientras que el efecto de perspectiva es muy poco acusado cuando D es del orden de varios miles. Puedes continuar ahora con el siguiente ejercicio para visualizar no uno, sino cuatro cubos al mismo tiempo en la pantalla.

OBJETOS MULTIPLES

Si conservas aún el programa del artículo anterior, te ahorrarás un montón de trabajo utilizando una simple instrucción de LOAD a fin de cargarlo de nuevo en tu ordenador. Si no lo guardaste, tendrás que teclearlo de nuevo. Asegúrate también de que tienes la *rutina Círculo,* que vimos en el primer artículo.

Como todo el programa está escrito en bloques o rutinas autónomos, resulta sencillo modificarlo para que dibuje figuras múltiples en vez de dibujar una sola. Por ejemplo, para dibujar cuatro cubos, no tienes más que especificar su posición en la pantalla y llamar a la *rutina Cubo* cuatro

veces. El mejor sitio para especificar la posición es la rutina de transformación de coordenadas. Modifica esta parte del listado de la manera siguiente, pero no ejecutes el programa todavía:

```
85ØØ X1=T1*X+T4*Y+T7+XO
851Ø Y1=T2*X+T5*Y+T8+YO
852Ø Z1=T3*X+T6*Y+T9+ZO
```

Las variables XO, YO y ZO al final de estas líneas especifican un desplazamiento en cada una de las tres direcciones de los ejes de coordenadas a fin de determinar la separación de los cuatro cubos. Estas variables son las que reciben valores justo antes de que se dibuje cada cubo.

Teclea ahora la siguiente sección del programa y ejecútalo para observar el efecto del desplazamiento:

```
12Ø L=2Ø:PP=2Ø:N=1
15Ø GOSUB15ØØ
15ØØ XO=−PP:YO=−PP:ZO
     =Ø
152Ø GOSUB1ØØØ
153Ø XO=PP:YO=−PP:ZO=Ø
154Ø GOSUB1ØØØ
155Ø XO=PP:YO=PP:ZO=Ø
156Ø GOSUB1ØØØ
157Ø XO=−PP:YO=PP:ZO=Ø
158Ø GOSUB1ØØØ
159Ø RETURN
```

Al ejecutar el programa, tienes que ir respondiendo a los mensajes que te aparecen en la pantalla. El primer mensaje te pide que introduzcas un valor para D, la distancia al plano de proyección. Un valor de 1000 es bueno para empezar. El siguiente mensaje te pide que introduzcas la posición del ojo, por lo que tendrás que teclear los valores de X, Y y Z. Aquí te presentamos un conjunto de valores que te darán determinadas imágenes en la pantalla. Prueba con ellos y observa cómo se distorsionan las imágenes cuando el valor de D es pequeño, y cómo con D grande la distorsión es inapreciable. Ensaya otros valores a tu gusto, incluyendo alguno de varios miles para D, así como valores negativos para X, Y y Z.

D	X	Y	Z
100	55	0	0
500	100	1	2
200	50	0	8

La rica variedad de vistas posibles con este programa te abre ya amplios horizontes para realizar experimentos interesantes, pero no tienes por qué ceñirte únicamente a cambiar los valores de la posición del ojo. Como habrás visto, la línea 120 de la última sección de programa que has tecleado define como variables la longitud de la arista del cubo (L), la separación de un cubo a otro (PP) y el número de cuadrados de la retícula (N) para cada una de las caras; aquí tienes pues otras tres variables que puedes modificar antes de hacer correr el programa.

DIBUJANDO UN GLOBO

Aunque el programa que acabas de introducir es corto, es capaz de configurar imágenes complejas, debido a que puede llamar a cuaquiera de las rutinas que componen el resto del listado. Una de estas rutinas podría ser la que dibuja círculos concéntricos, y que ya comentamos en el artículo anterior. Si tu programa no contiene esta rutina, tecléala ahora y guarda una copia del programa completo para poder utilizarla en el futuro.

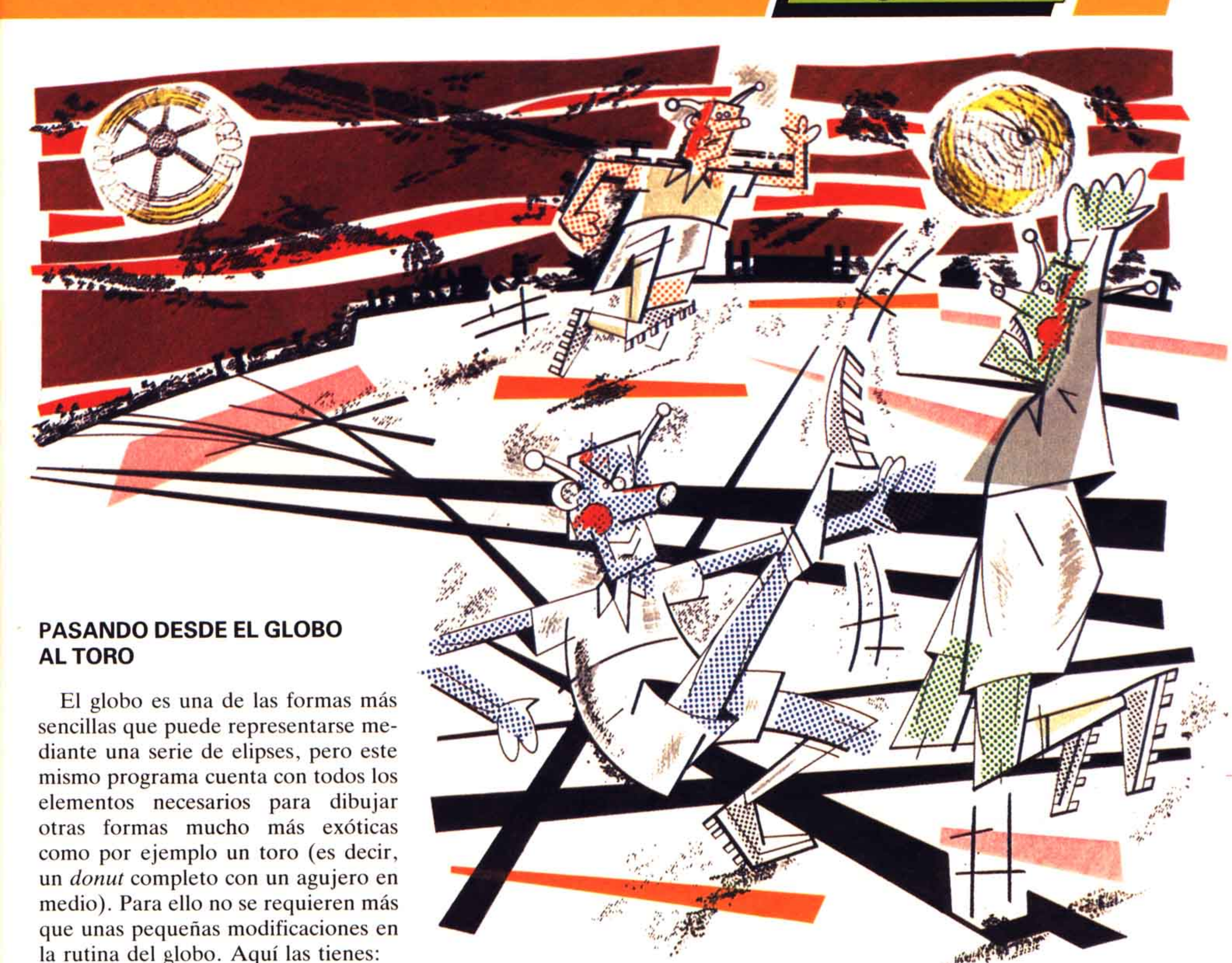
Ahora puedes modificar el programa para que en vez de llamar a las *rutinas del cubo, retícula y cara,* llame a la *rutina Círculo.* Borra la línea 120 y a continuación teclea las siguientes líneas:

```
1Ø1 XO=Ø:YO=Ø:ZO=Ø
15Ø R=2Ø:N=18:GOSUB2ØØØ
2ØØØ T7=Ø:T8=Ø:T9=Ø
2Ø1Ø T4=Ø:T5=Ø:T6=1
2Ø4Ø KA=2*PI/N
2Ø5Ø KB=Ø
2Ø6Ø FORK=1TOINT(N/2)
2Ø7Ø T1=COS(KB):T2=
     SIN(KB):T3=Ø
2Ø8Ø XS=Ø:YS=Ø:GO
     SUB6ØØØ
2Ø9Ø KB=KB+KA
21ØØ NEXTK
211Ø T1=1:T2=Ø:T3
     =Ø
212Ø T4=Ø:T5=1:T6=Ø
213Ø KA=KA/2
214Ø KB=Ø:RR=R
215Ø FORK=1TON
216Ø T7=Ø:T8=Ø:
     T9=RR*COS(KB)
217Ø XS=Ø:YS=
     Ø:R=RR*SIN(KB)
     :GOSUB6ØØØ
218Ø KB=KB+KA
219Ø NEXTK
22ØØ RETURN
```

La primera parte de la fase de ejecución de este programa te resultará familiar, ya que se trata de seleccionar un valor de D y a continuación un valor para la posición del ojo. Una vez introducidos estos valores, la línea 150 llama a la rutina recién tecleada para dibujar la imagen de un globo. Esto se hace a partir de una serie de elipses (o círculos, dependiendo de la posición del ojo). Por una parte están las líneas de longitud (que pasan por los polos de la esfera) y por otra las de latitud, que son paralelas al ecuador.

El valor del radio de la esfera está dado por la variable R (línea 150), mientras que el número de líneas de longitud y latitud está dado por N, que tiene que ser un número mayor que 2. Las líneas 2000 a 2050 definen las variables para posicionar el globo y se utilizan además como contadores. El bucle que hay entre las líneas 2060 y 2100 se ocupa de dibujar las líneas de longitud (los meridianos), mientras que el que va desde la línea 2150 a la 2190, dibuja las líneas de latitud constante (los paralelos).

El eje de la esfera es la línea que pasa por los polos y su dirección depende de los valores que hayas introducido. Para ver la forma en que quedan situados los globos en relación con los ejes de la pantalla, teclea el valor 0 para la posición de dos de las coordenadas del punto de vista, por ejemplo (0,0,200) y observa el resultado. Prueba ahora a introducir valores positivos o negativos para las tres coordenadas, para ver cómo se desplazan los polos respecto a los ejes.



PASANDO DESDE EL GLOBO AL TORO

El globo es una de las formas más sencillas que puede representarse mediante una serie de elipses, pero este mismo programa cuenta con todos los elementos necesarios para dibujar otras formas mucho más exóticas como por ejemplo un toro (es decir, un *donut* completo con un agujero en medio). Para ello no se requieren más que unas pequeñas modificaciones en la rutina del globo. Aquí las tienes:

```
150 R=10:N=18:RT=20:
    N2=18:GOSUB2000
2040 KA=2*PI/N2:NC=N2
2045 IFRT=0THENNC=INT(NC/
     2)
2060 FORK=1TONC
2080 XS=RT:YS=0:
     GOSUB6000
2130 KA=2*PI/N
2135 IFRT=0THENKA
     =KA/2
2170 XS=0:YS=0:R=
     RT+RR*SIN(KB)N=N2:GO
     SUB6000
```

Ejecuta el programa y como respuesta a los mensajes de pantalla teclea unos valores adecuados para D y para la posición del ojo. Esta vez la variable R corresponde al radio interno del tubo, mientras que el número de segmentos radiales está dado por N. Estos segmentos corresponden a las líneas longitudinales del globo. RT asigna la distancia (un radio) desde el origen al centro del tubo, mientras que N2 da el número de segmentos a lo largo de la circunferencia del tubo, que corresponden a las latitudes de la esfera. El valor de RT debe ser mayor o igual que R, aunque de hecho la rutina sigue funcionando bien aunque no ocurra esto. Modifica el valor de estas variables para observar el efecto resultante. Cuando RT es cero y N es igual que N2, el toro degenera en una esfera, tal como aparecía dibujada en la *rutina Globo*.

COMBINACION DE IMAGENES

En el programa anterior que dibujaba cuatro imágenes juntas en la pantalla, te resultará muy sencillo sustituir la *rutina Cubo* por la *rutina Globo* o la *rutina Toro*. De una forma igualmente sencilla, puedes utilizar una combinación cualquiera de las tres rutinas. Por ejemplo, podrías dibujar un cubo dentro de una esfera o de un toro, o incluso un cubo dentro de una esfera que a su vez está dentro de un toro. También puedes insertar sentencias GOTO en los puntos adecuados del programa para evitar que se dibujen las últimas caras de modo que la imagen no resulte muy recargada. Teclea las siguientes líneas y obten-

P y R

¿Hay alguna forma sencilla de acelerar el dibujo de la estación espacial?

La forma más radical de acelerar el programa consiste en reducir el número de pasos de las rutinas básicas: la *rutina Círculo* para la estación espacial y la *rutina Cubo* para el caso de los cubos. Sin embargo, este método reducirá la suavidad de las elipses y estropeará la forma de las estructuras. Un método mejor es reducir el número de veces que se llama a las rutinas.

Podrías tener por ejemplo menos secciones transversales y longitudinales del toro, reduciendo los valores de R, N, RT y N2 en la rutina del toro que empieza en la línea 2000. Esto implica obviamente una modificación del programa para que la rutina sea llamada de forma que espacie con regularidad los círculos o las elipses y los saltos después de la última serie de curvas, en vez de seguir con los movimientos del dibujo más de lo necesario.

Probablemente la forma más sencilla de acelerar la ejecución consiste en saltarse el dibujo de determinadas secciones (tales como los radios de la estación). Esto se hace de forma sencilla con unos cuantos GOTO estratégicamente distribuidos. Obtendrás una impresionante representación de una estación espacial:

```
150 GOSUB3000
3000 R=6:N=24:RT=40:
     N2=24:GOSUB2000
3030 FORF=1TO6
3040 A=(F+.25)*PI/3:L1=10:
     L2=34:N1=12:R=2:
     N2=8:GOSUB3500
3050 NEXTF
3060 R=10:N=12:RT=0:
     N2=12:GOSUB2000
3070 RETURN
3500 SA=SIN(A)
3530 CA=COS(A)
3540 T1=-SA:T2=CA:
     T3=0
3550 T4=0:T5=0:T6=1
3560 EA=(L2-L1)/N1
3570 EB=L1
3580 FORE=0TON1
3590 T7=EB*CA:T8=EB*
     SA:T9=0
3600 XS=0:YS=0:
     N=N2:GOSUB6000
3610 EB=EB+EA
3620 NEXTE
3630 T1=CA:T2=SA:T3=0
3640 FORE=1TON2
3650 EA=E*PI/8
3660 T7=R*SA*COS(EA):T8=
     -R*CA*COS(EA):
     T9=R*SIN(EA)
3670 XS=L1:YS=
     0:XE=L2:
     YE=0:GOSUB9500
3680 NEXTE
3690 RETURN
```

Este programa dibuja un toro con seis radios apuntando hacia el centro, en el que se encuentra una esfera. La imagen resultante se parece a determinados modelos de estación espacial.

El campo de experimentación que te permite el programa del toro es virtualmente ilimitado. Ensaya algunos valores pequeños de D (por ejemplo entre 100 y 500) junto con diversos valores de X, Y y Z para obtener interesantes vistas en perspectiva, especialmente al recortar el segmento más corto del toro. Prueba también con valores altos de D para observar las distorsiones que resultan.

Ahora ya es el momento de pasar a los toques finales, tales como el color del fondo y el de la tinta.

MEJORANDO LA CLARIDAD

Cuando las imágenes de los cuatro cubos son pequeñas, como sucede cuando los valores de X, Y y Z correspondientes a la posición del ojo son mucho mayores que la distancia de observación (D), es difícil ver lo que sucede cuando el programa avanza durante la fase del dibujo. Puedes mejorar la claridad insertando sentencias GOTO en lugares adecuados del programa entre las líneas 1000 y 1170. La más sencilla es GOTO 1220, insertada en la línea 1125 (que es una nueva línea) para el caso del programa de los cuatro cubos.

En el sorteo correspondiente al número 16 entre quienes escribisteis mandando vuestros votos a los MEJORES DE INPUT han resultado ganadores:

Nombre	Localidad	Juego elegido
Eddie Day Leal	Mendoza (Argentina)	Who Dares Wins II
Juan Manuel Cruz Navarro	Zaragoza	Army Moves
José Pedro de la Cruz Domínguez	Miravalles (Bizkaia)	Némesis
Carlos Quiza Fernández	Madrid	Penguin Adventure
Luis Dalmau Schilt	Barcelona	Vampire Killer
César Toquero Zamora	Valladolid	Green Beret
Fco. Javier Cardiel Moreno	Parcela (Zaragoza)	Livingstone
Mertxe Hernández Maraña	San Sebastián (Guipúzcoa)	Némesis
Marcelino Hermida López	Maniños (La Coruña)	Penguin Adventure
Roger Palau Capdevila	Barcelona	Army Moves

DETECCION DE OBJETOS EN LA PANTALLA

- DETECCION DE FORMAS EXTRAÑAS
- COMPROBACIONES DE COLOR
- EL COMANDO ATTR
- LA PELOTA QUE REBOTA
- DETECCION DE LOS CHOQUES

Tu ordenador puede seguir a los gráficos utilizando determinados comandos que le permiten observar su propia pantalla. Esto resulta especialmente útil en los juegos en los que se producen colisiones.

En las situaciones en que hayas creado una figura detallada sobre la pantalla, ¿cómo puedes asegurarte de que el siguiente objeto que añadas a la imagen no coincidirá con algo que ya había allí antes?

Una respuesta obvia es que seas cuidadoso y sistemático al llevar la cuenta de las zonas de la pantalla que ya han sido utilizadas. Pero esto puede llegar a resultar enormemente complicado, y hay circunstancias en que puede ser casi imposible, como es el caso de los gráficos móviles. Y, sin embargo, éste es precisamente el tipo de problemas con el que te encuentras al escribir programas de juegos y cosas parecidas en las que hay invasores y vehículos espaciales moviéndose por una galaxia, o fantasmas recorriendo un laberinto. En tales casos tienes que asegurarte de que no se impriman dos gráficos en el mismo sitio, o de qué sucederá si se produce la colisión y, por ejemplo, se desintegre la nave espacial.

DETECCION DE FORMAS EXTRAÑAS

Supongamos por ejemplo que deseas escribir un programa que haga aparecer en pantalla una pelota dando botes. En este caso la cosa es muy sencilla: como conoces las coordenadas de los cuatro lados de la pantalla, puedes incluir en tu programa cuatro condiciones IF ... THEN para comprobar si la pelota ha llegado o no a uno de los cuatro lados. Ello se hace comparando las coordenadas de la pelota con las coordenadas conocidas de los lados. Pero ¿qué sucede cuando lo que quieres comprobar es si la pelota ha golpeado contra un objeto de forma más extraña, por ejemplo circular?

Podrías utilizar el mismo método, disponiendo unas cuantas comprobaciones del tipo IF ... THEN que contengan los detalles de las coordenadas del círculo para ver en qué puntos golpea la pelota con los lados. Pero debido a que la forma curva es especialmente compleja, se necesitaría un númer de comprobaciones muy grande. Y ya sabes que el ordenador necesita mucho tiempo para ejecutar cada comprobación IF ... THEN. Un programa escrito de esta forma resultaría extremadamente lento, y el movimiento resultante sobre la pantalla se produciría a base de tirones o sacudidas.

Existe un límite para las velocidades que es posible obtener utilizando el BASIC. Pero tu ordenador tiene un comando que te permite detectar la presencia de cualquier objeto sobre la pantalla con mucha mayor rapidez que la comprobación de las coordenadas. Esto es posible aunque no se conozca su posición.

EL COLOR POR MEDIO DE NUMEROS

El comando POINT de MSX te devuelve como respuesta el color especificado de cada punto. Esto significa que puedes detectar la presencia de cualquier objeto, sin más que especificar su color y comprobando todas las posiciones de pantalla. Así, en el ejemplo anterior, con un círculo rojo resultaría el número de código del color rojo cada vez que se presentara. Comprobando únicamente este código el programa podría hacer que la pelota rebotara hacia afuera del círculo, o cualquier otra cosa.

En cuanto a la sintaxis de este comando, POINT ha de ir seguido de las coordenadas de pantalla del punto cuyo color quieres averiguar, colocadas entre paréntesis. Podrías escribir, pues, algo como esto:

```
X=POINT(1Ø,1Ø)
```

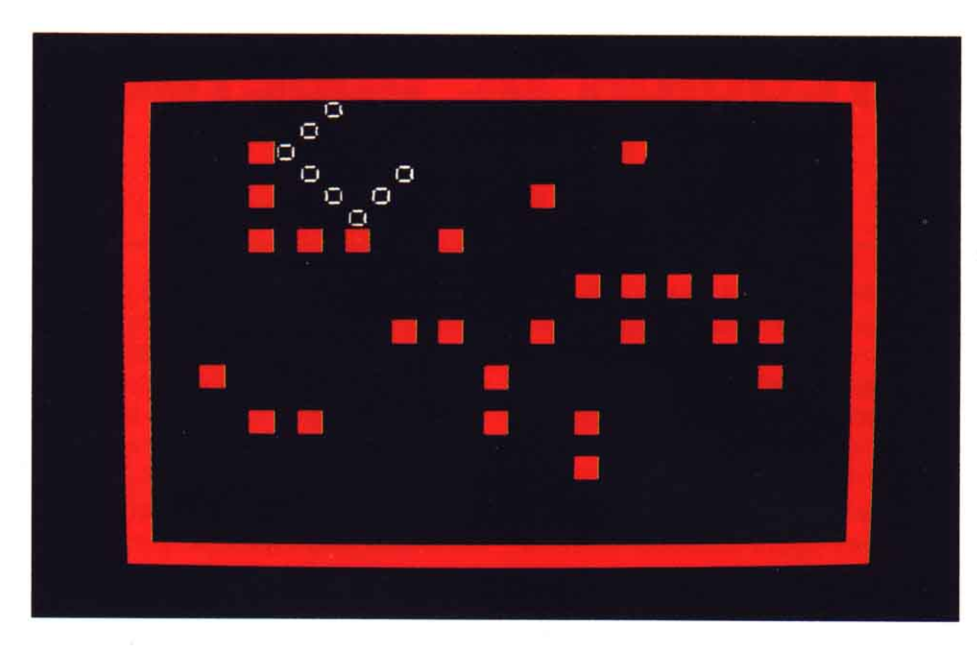
En este caso X contiene el código de color (10,10) de la pantalla. Teclea y ejecuta ahora el siguiente programa que es un ejemplo de utilización del comando POINT.

```
1Ø OPEN "grp:" AS #1
2Ø COLOR 8,1,1
3Ø SCREEN 2
4Ø FOR T=Ø TO 255
5Ø VPOKE T+8192,2ØØ
6Ø NEXT T
7Ø FOR T=6143 TO 6143-255
   STEP -1
8Ø VPOKE T+8192,2ØØ
9Ø NEXT T
1ØØ FOR T=Ø TO 5888 STEP
    256-8
11Ø FOR V=T TO T+7
12Ø VPOKE 8192+V,2ØØ
13Ø NEXT V
14Ø T=T+8
15Ø NEX T
16Ø FOR T=248 TO 6143 STEP
    256-8
17Ø FOR V=T TO T+7
18Ø VPOKE 8192+V,2ØØ
19Ø NEXT V
2ØØ T=T+8
21Ø NEXT T
22Ø X=INT(RND(-TIME)*256)
23Ø Y=INT(RND(1)*191)
24Ø Z=Z+1
25Ø IF Z=25 THEN 29Ø
26Ø PSET (X,Y),Ø
27Ø PRINT #1,"o"
28Ø GOTO 22Ø
29Ø A1$=CHR$(&BØØ1111ØØ)
```

```
3ØØ A2$=CHR$(&BØ111111Ø)
31Ø A3$=CHR$(&B1ØØ11ØØ1)
32Ø A4$=CHR$(&B11111111)
33Ø A5$=CHR$(&B111ØØ111)
34Ø A6$=CHR$(&B1Ø1111Ø1)
35Ø A7$=CHR$(&BØ1ØØØØ1Ø)
36Ø A8$=CHR$(&BØØ1111ØØ)
37Ø A$=A1$+A2$+A3$+A4$+
    A5$+A6$+A7$+A8$
38Ø Z1=1:Z2=1
39Ø SPRITE$(Ø)=A$:
    X=2Ø:Y=2Ø
4ØØ PUT SPRITE Ø,(X,Y),3,Ø
41Ø IF POINT (X+8,Y+4)=8
    THEN GOSUB 47Ø
42Ø IF POINT (X-1,Y+4)=8
    THEN GOSUB 5ØØ
43Ø IF POINT (X+4,Y+8)=8
    THEN GOSUB 53Ø
44Ø IF POINT (X+4,Y-1)=8
    THEN GOSUB 56Ø
45Ø GOSUB 59Ø
46Ø GOTO 4ØØ
47Ø REM 1
48Ø X=X-1:Z1=-1:Z2=Z2:
    PLAY"v15163o8dd"
49Ø RETURN
5ØØ REM 2
51Ø X=X+1:Z1=1:Z1=Z2:
    PLAT"v15163o8dd"
52Ø RETURN
53Ø REM 3
54Ø Z1=Z1:Z2=-1:PLAY
    "v15163o8dd"
55Ø RETURN
56Ø REM 4
57Ø Z1=Z1:Z2=1:
    PLAY"v15163o8dd"
58Ø RETURN
59Ø REM Ø
6ØØ X=X+Z1:Y=Y+Z2
61Ø RETURN
```

La imagen que aparece en el MSX consta de una serie de espacios rojos presentados aleatoriamente dentro de un recinto. Al hacer correr el programa, el ordenador empieza a mover la pelota con una dirección inicial aleatoria.

Cuando la pelota llega al borde de uno de los bloques, rebota y toma una dirección diferente.



DETECCION DE LAS COLISIONES

Las comprobaciones para detectar el instante en que la pelota choca con el borde de uno de los cuadrados, se hacen por medio del comando POINT. Como los cuadrados se imprimen aleatoriamente, la única alternativa para el seguimiento sería asignar una variable para la coordenada x y otra para la coordenada y de cada cuadro.

De esta forma, al poder utilizar dos variables para cada cuadrado, aparte del hecho de utilizar una gran cantidad de memoria, se necesitaría una cantidad enorme de comprobaciones IF ... THEN.

Estas comprobaciones significarían que lo que ahora es un programa razonablemente rápido, se convirtiera en un programa insoportablemente lento.

Sería muy sencillo para ti utilizar comprobaciones del tipo IF ... THEN para detectar las colisiones de la pelota con los bordes, ya que sus coordenadas se definen con mucha facilidad.

Pero una vez que hayas utilizado el comando POINT para comprobar si la pelota ha chocado o no contra un bloque rojo lo suyo es utilizarlo también para las colisiones con los bordes, especialmente en este caso en que el color es el mismo que el de los bloques.

La línea 10 abre un fichero de gráficos para que podamos escribir caracteres en una pantalla de gráficos, la línea 20 prepara el color de la pantalla, la línea 30 inicializa la pantalla en modo de screen 2, desde la línea 40 hasta la 200 se diseña el borde rojo de la pantalla.

El bucle situado desde la línea 220 hasta la 280 sitúa aleatoriamente 25 bloques de color rojo, dispersos por la pantalla, desde la línea 290 hasta la 370 se define un esprite con forma de cara.

Desde la línea 410 hasta la línea 440 se lee el código de color de los lados del esprite.

Las líneas siguientes, hasta el final del listado, son rutinas que se encargan de desviar el movimiento del esprite, según el lado por el que se haya producido un choque.

NO OLVIDES EL TELEFONO... ☎

Cuando, por cualquier motivo, nos escribas, no olvides indicar tu número de teléfono. Así nos será más fácil y rápido ponernos en contacto contigo. Gracias.



PUESTO	TITULO	PORCENTAJE
1.º	NEMESIS	21,8 %
2.º	SOCCER	18,6 %
3.º	KNIGHTMARE	13 %
4.º	GREEN BERET	13 %
5.º	THE GOONIES	7,2 %
6.º	DUSTIN	6 %
7.º	VAMPIRE KILLER	6 %
8.º	LIVINGSTONE	5,3 %
9.º	SPIRITS	5,1 %
10.º	PROFANATION	4 %
		100,0 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «LOS MEJORES INPUT».

ELIGE TUS PROGRAMAS

Hemos pensado que es interesante disponer de un *ranking* que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Aribau, 185. Planta 1. 08021 Barcelona — MSX N.º 17

1.er Título elegido
2.º Título elegido
3.er Título elegido
Programa que te gustaría conseguir
Qué ordenador tienes
Nombre
1.er Apellido
2.º Apellido
Fecha de nacimiento
Teléfono
Dirección
Localidad
Prov.

CURSOR EXTERIOR PARA ORDENADORES MSX

Todos nos hemos sentido tentados alguna vez de aporrear despiadadamente a nuestro compañero de fatigas en las luchas galáctico-infernales después de haber perdido la última vida que nos quedaba en la batalla por la supremacía del Universo. Pero siempre refrenábamos nuestro impulso «joystickcida» pensando en mejores tiempos y en la batalla siguiente...

Los joysticks, aparte de ser objeto de nuestros más ruines pataleos, son unos perfectos desconocidos, a pesar de tenerlos siempre al alcance de la mano.

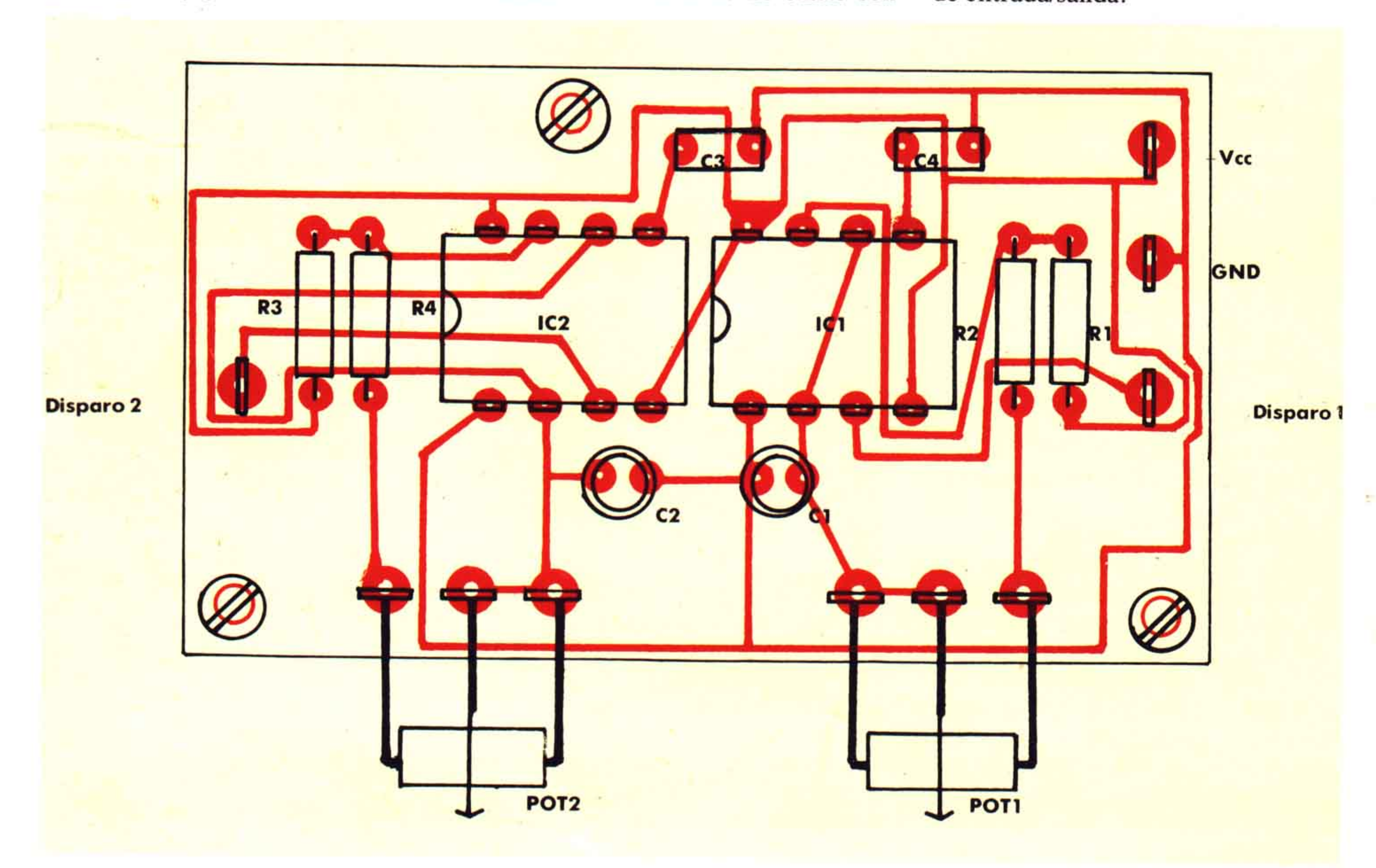
En el presente artículo trataremos de acercanos, conocer y hasta poder construir un joystick con las variantes más mortíferas ideadas hasta el momento: disparo programable, ráfagas atómicas a base de **tontolio frito** y servosensibilidad en el uso de los controles.

FUNCIONAMIENTO

Todos los joysticks están compuestos en base a unos pequeños interruptores que generan, en las líneas del port de entrada del joystick, unas señales lógicas que corresponden a ceros y unos. Dichas señales, el port las transforma en información inteligible para el microprocesador: esta conversión se denomina **codificación.**

Los niveles lógicos en los joysticks tienen valor 1 cuando no existe contacto entre la línea y masa, y valor 0 cuando hay contacto entre ellas.

Para que se realice la codificación, el **controlador de vídeo** (VDP) tiene incluida dicha opción codificadora, donde las señales provenientes del joystick son transformadas y ubicadas en unos determinados registros que dispone el VDP. Dichos registros serán leídos por el **BIOS** (Basic Input Output System) y transmitidos al microprocesador para que trabaje con ellos. Esta tarea se repite en otros casos, como pueden ser lectura del teclado, lectura/escritura en cassette o disco, o escritura en la VRAM. La comunicación en estos casos se realiza a través de registros contenidos en el mapa de entrada/salida.



- EL JOYSTICK: UN DESCONOCIDO
- LA CODIFICACION Y EL CONTROLADOR DE VIDEO
- MATERIALES
- MONTAJE DEL CURSOR EXTERNO
- UTILIZACION DE CABLES DE DIFERENTES COLORES
- MANDO MANUAL O MODO AUTOMATICO
- TEST DE PRUEBA

El Z-80 leerá los datos de los registros por diferentes métodos, como, por ejemplo, el de búsqueda periódica de datos *(pooling)* o mediante interrupciones.

MATERIALES

Los materiales que emplearemos se pueden dividir en dos grupos:
- materiales para la construcción básica de un joystick tales como:
  - una caja de 18*10*3 cm;
  - seis pulsadores con sus correspondientes teclas;
  - 1,5 m de cable de 8 o más hilos;
  - un conector para joystick de 9 pines; y
- materiales para la obtención del disparo programable:
  - dos conmutadores de un circuito y dos posiciones;
  - dos potenciómetros de 50 kilo-ohmios lineales;
  - dos circuitos integrados 555;
  - un LED rojo de 3 o 5 mm;
  - un LED verde de 3 o 5 mm;
  - dos condensadores de 10 microfaradios y 25 voltios;
  - dos condensadores de 100 nanofaradios;
  - cuatro resistencias de 4.700 ohmios, de tolerancia 5 % y potencia 0,25 watios;
  - dos resistencias de 1.000 ohmios, de tolerancia 5 % y potencia 0,25 watios;
  - un zócalo de 16 pines;
  - una plaquita de *veroboard* (circuito impreso prefabricado) de aproximadamente 80*50 mm.

Realizando un balance del costo de los componentes, se podría decir que el primer bloque de materiales será aproximadamente de unas 1.500 ptas., y el segundo se elevará en 500 ptas., lo que nos dará un total de 2.000 ptas.

Los pulsadores van a ser los componentes que van a recibir un mayor desgaste; por ello es aconsejable la elección de materiales de calidad, de lo contrario podríamos encontrarnos con el problema de tenerlos que sustituir con bastante frecuencia (que a la larga haría subir el costo de fabricación).

MONTAJE DEL CURSOR EXTERNO

Lo primero que haremos será mecanizar la caja donde se alojará el cursor, aunque lo más normal es adaptarlo a las dimensiones reales de la caja. Igualmente ha de mantenerse una distancia en diagonal (de las teclas del cursor) no superior a 50 mm, atendiendo a razones de ergonomía.

Cuando se hallen instalados todos los componentes en la caja (pulsadores, potenciómetros e interruptores), realizaremos el cableado de los mismos, teniendo en cuenta que a cada pulsador le corresponda un hilo del conector (de joystick) y otro común a todos: el de masa (que se conectará también al terminal de masa del conector).

Para facilitar nuestra tarea y no enrollarnos (nunca mejor dicho) con los hilos, se aconseja el uso de cables con diferentes colores para una más fácil identificación a la hora del montaje.

En el caso de los pulsadores de disparo esto será diferente, pues nos interesa la opción de disparo automático, y si uniéramos directamente el pulsador a masa sólo obtendríamos disparo manual. Utilizando un conmutador seleccionaremos mando manual (conectando los pulsadores a masa) o modo automático (conectándolos a la salida del oscilador).

El montaje del oscilador se puede realizar de diferentes formas: sobre una placa de circuito impreso prefabricado de tiras de cobre o de arandelas (con un paso de décima de pulgada) o más elegantemente (el que utili-



zamos en nuestro cursor) mediante un circuito impreso con distribución de pistas.

Para realizar este último c. i. podemos adoptar diferentes métodos, desde el más sencillo, como es el de dibujar directamente en una placa de fibra de vidrio o baquelita la distribución de las pistas, hasta el uso de calcas o transferibles (que se venden en las tiendas de componentes electrónicos para tal fin).

La placa del c. i. deberá ser taladrada en los lugares donde vayan a ir alojados los terminales de los componentes, siendo el diámetro del taladro proporcional al diámetro de los terminales a conectar. Una vez hecho esto, soldaremos los componentes aconsejando que el c. i. se conecte a través de terminales, como pueden ser espadines.

Cabe señalar que todo el montaje anterior sólo es necesario cuando deseemos que el cursor posea disparo semiautomático, de lo contrario solamente conectando los pulsadores de disparo a masa sería suficiente, obteniendo disparo de tipo manual.

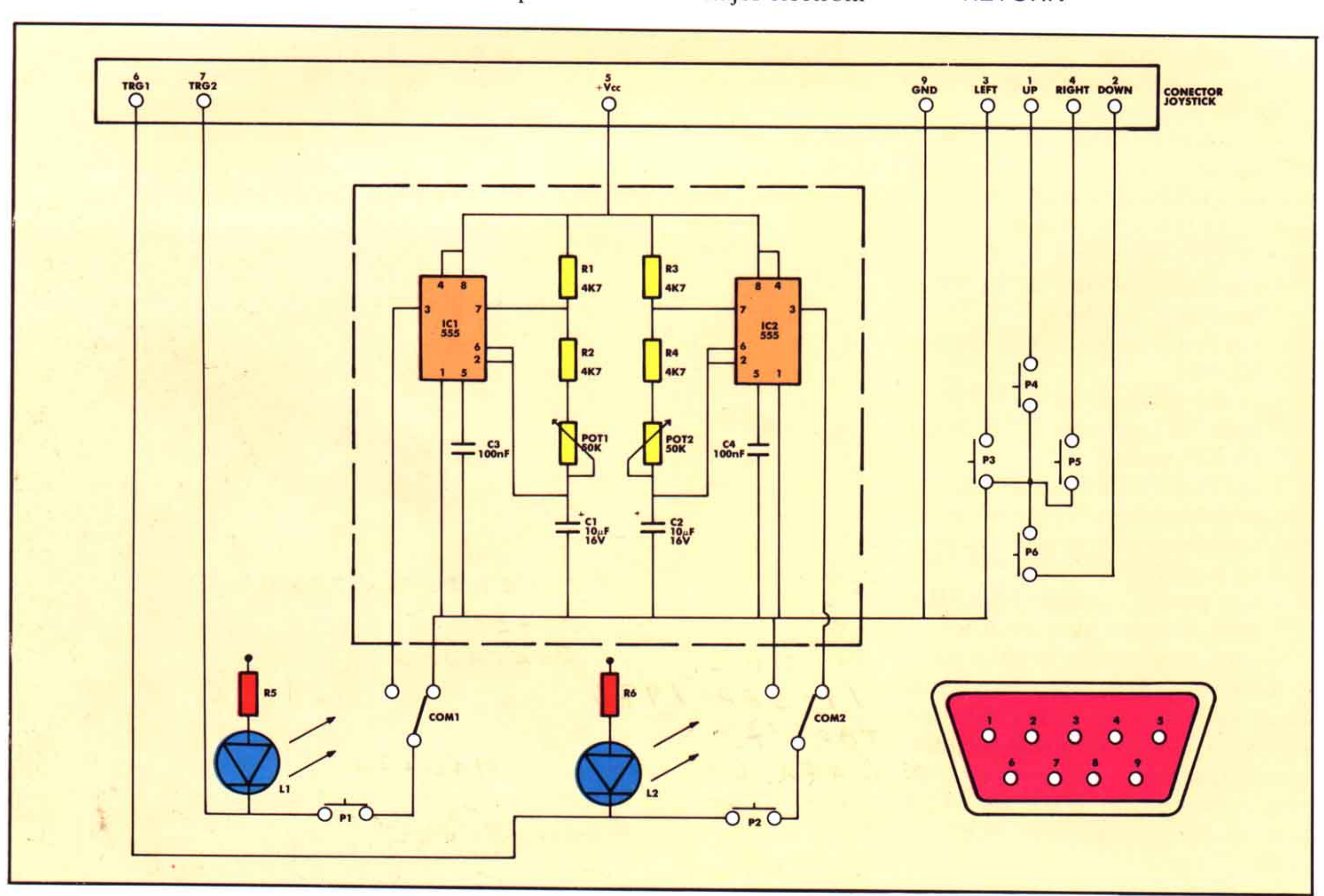
Los leds incluidos en el esquema indicarán el momento exacto en que se realiza el disparo y si la velocidad de éste es aceptada por el programa en memoria, ya que no todos los masacramarcianos leen los disparadores de los joysticks a la misma velocidad. En muchos programas el disparo continuo es interpretado como vacío absoluto de disparo.

Si se siguen todas las indicaciones correctamente y se tiene cuidado a la hora del montaje de no realizar ningún tipo de cruce, deberíamos llevar a feliz término la realización del cursor. De todas formas, y cara a la seguridad del ordenador, es conveniente verificar que las líneas de alimentación (de 5 voltios) y la masa no estén cruzadas, ya que peligraría la fuente de alimentación del ordenador.

Para aquellos que tengan algún tipo de experiencia en montajes electrónicos, indicamos que existe una gran cantidad de variaciones que se podrían hacer sobre este montaje, donde la única limitación es la imaginación de cada uno, quedando abierta la invitación a posibles mejoras sobre el aparato.

TEST DE PRUEBA

```
10 CLS
20 A=STICK(1):LOCATE 10,15
30 ON A GOSUB 50,60,70,80,90,
   100,110,120
40 GOTO 20
50 PRINT" ARRIBA ":RETURN
60 PRINT"DIAG.DER.SUP.":RETURN
70 PRINT" DERECHA ":RETURN
80 PRINT"DIAG.DER.INF.":RETURN
90 PRINT" ABAJO ":RETURN
100 PRINT"DIAG.IZQ.INF.":
    RETURN
110 PRINT" IZQUIERDA ":
    RETURN
120 PRINT"DIAG.IZQ.SUP.":
    RETURN
```



MOVIENDO SPRITES EN CODIGO MAQUINA

- MOVIMIENTO EN PANTALLA
- SPRITES EN C.M.
- LAS RUTINAS DE LA ROM
- EL BIOS
- CARGADORES EN BASIC

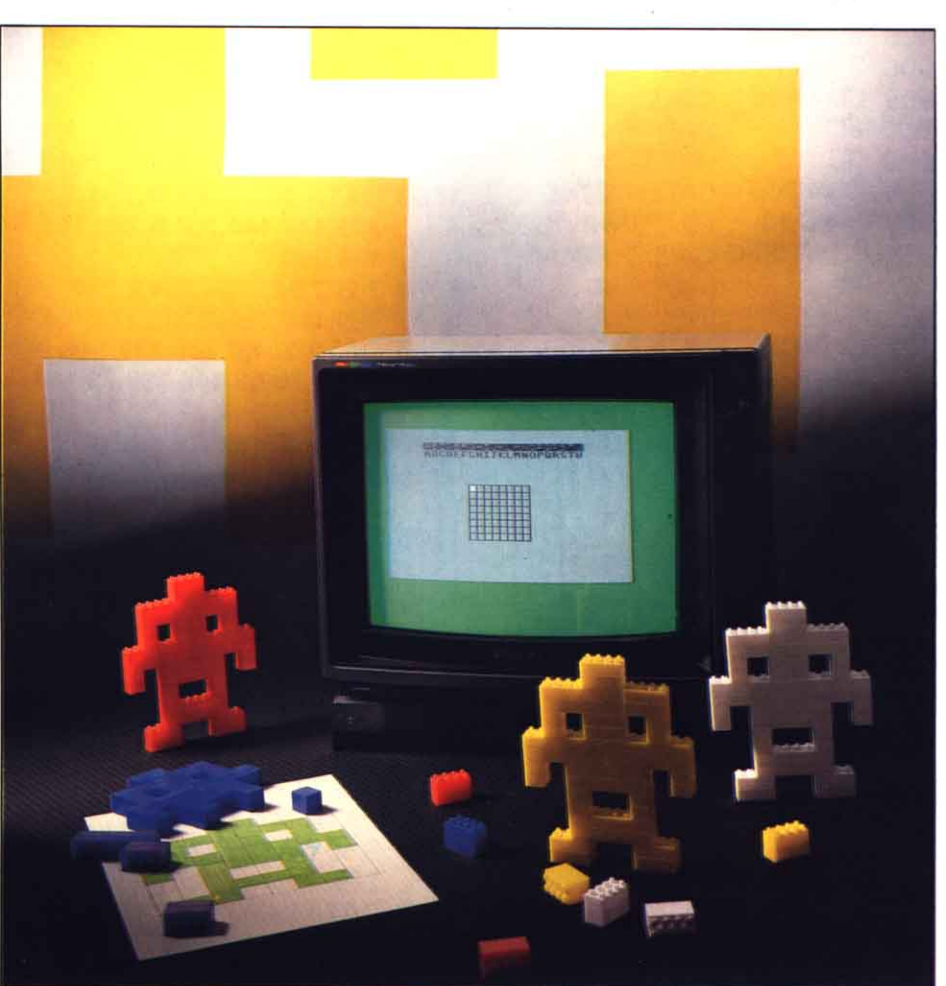
El programa que presentamos este mes ha sido escrito íntegramente en código máquina. Como es costumbre en estos casos, hemos definido también unos cargadores en BASIC para que los que no disponéis de un programa ensamblador podáis ejecutar el programa.

Éste, en sí, consta de dos partes, la primera se compone de la rutina de movimiento del sprite, y la segunda está compuesta por los datos que conforman éste.

El programa se ha diseñado utilizando ampliamente las rutinas del BIOS (Basic Input Output System).

Teclea ahora el programa en tu ensamblador, o utiliza el cargador BASIC.

```
8FFE  ØEØ1      LD    C,Ø1
9ØØØ  Ø6F2      LD    B,F2
9ØØ2  CD47ØØ    CALL  ØØ47Ø
9ØØ5  3EØF      LD    A,ØF
9ØØ7  32E9F3    LD    (F3E9),A
9ØØA  3EØ1      LD    A,Ø1
9ØØC  32EAF3    LD    (F3EA),A
9ØØF  32EBF3    LD    (F3EB),A
9Ø12  CD62ØØ    CALL  ØØ62
9Ø15  CD72ØØ    CALL  ØØ72
9Ø18  21ØØ38    LD    HL,38ØØ
9Ø1B  11ØØAØ    LD    DE,AØØØ
9Ø1E  1A        LD    A,(DE)
9Ø1F  CD4DØØ    CALL  ØØ4D
9Ø22  23        INC   HL
9Ø23  13        INC   DE
9Ø24  FEAA      CP    AA
9Ø26  C21E9Ø    JP    NZ,9Ø1E
9Ø29  21ØC1B    LD    HL.1BØC
9Ø2C  3E64      LD    A,64
9Ø2E  CD4DØØ    CALL  ØØ4D
9Ø31  21ØD1B    LD    HL,1BØD
9Ø34  3E64      LD    A,64
9Ø36  CD4DØØ    CALL  ØØ4D
9Ø39  21ØE1B    LD    HL,1BØE
9Ø3C  3EØ3      LD    A,Ø3
9Ø3E  CD4DØØ    CALL  ØØ4D
9Ø41  21ØF1B    LD    HL,1BØF
9Ø44  3EØ7      LD    A,Ø7
9Ø46  CD4DØØ    CALL  ØØ4D
9Ø49  76        HALT
9Ø4A  3EØØ      LD    A,ØØ
9Ø4C  CDD8ØØ    CALL  ØØD8
9Ø4F  FEFF      CP    FF
9Ø51  CA2191    JP    Z,9121
9Ø54  ØØ        NOP
9Ø55  3EØØ      LD    A,ØØ
9Ø57  CDD5ØØ    CALL  ØØD5
9Ø5A  FEØ1      CP    Ø1
9Ø5C  CA859Ø    JP    Z,9Ø85
9Ø5F  FEØ3      CP    Ø3
9Ø61  CA939Ø    JP    Z,9Ø93
9Ø64  FEØ5      CP    Ø5
9Ø66  CAA19Ø    JP    Z,9ØA1
9Ø69  FEØ7      CP    Ø7
9Ø68  CAAF9Ø    JP    Z,9ØAF
9Ø6E  FEØ2      CP    Ø2
9Ø7Ø  CABD9Ø    JP    Z,9ØBD
9Ø73  FEØ4      CP    Ø4
9Ø75  CAD69Ø    JP    Z,9ØD6
9Ø78  FEØ6      CP    Ø6
9Ø7A  CAEF9Ø    JP    Z,9ØEF
9Ø7D  FEØ8      CP    Ø8
9Ø7F  CAØ891    JP    Z,91Ø8
9Ø82  C3499Ø    JP    9Ø49
9Ø85  21ØC1B    LD    HL,1BØC
9Ø88  CD4AØØ    CALL  ØØ4A
9Ø8B  3D        DEC   A
9Ø8C  3D        DEC   A
9Ø8D  CD4DØØ    CALL  ØØ4D
9Ø9Ø  C3499Ø    JP    9Ø49
9Ø93  21ØD1B    LD    HL,1BØD
9Ø96  CD4AØØ    CALL  ØØ4A
```



```
9Ø99  3C         INC    A
9Ø9A  3C         INC    A
9Ø9B  CD 4D ØØ   CALL   ØØ4D
9Ø9E  C3 49 9Ø   JP     9Ø49
9ØA1  21 ØC 1B   LD     HL,1BØC
9ØA4  CD 4A ØØ   CALL   ØØ4A
9ØA7  3C         INC    A
9ØA8  3C         INC    A
9ØA9  CD 4D ØØ   CALL   ØØ4D
9ØAC  C3 49 9Ø   JP     9Ø49
9ØAF  21 ØD 1B   LD     HL,1BØD
9ØB2  CD 4A ØØ   CALL   ØØ4A
9ØB5  3D         DEC    A
9ØB6  3D         DEC    A
9ØB7  CD 4D ØØ   CALL   ØØ4D
9ØBA  C3 49 9Ø   JP     9Ø49
9ØBD  21 ØC 1B   LD     HL,1BØC
9ØCØ  CD 4A ØØ   CALL   ØØ4A
9ØC3  3D         DEC    A
9ØØC  4 3D       DEC    A
9ØC5  CD 4D ØØ   CALL   ØØ4D
9ØC8  21 ØD 1B   LD     HL,1BØD
9ØCB  CD 4A ØØ   CALL   ØØ4A
9ØCE  3C         INC    A
9ØCF  3C         INC    A
9ØDØ  CD 4D ØØ   CALL   ØØ4D
9ØD3  C3 49 9Ø   JP     9Ø49
9ØD6  21 ØC 1B   LD     HL,1BØC
9ØD9  CD 4A ØØ   CALL   ØØ4A
9ØDC  3C         INC    A
9ØDD  3C         INC    A
9ØDE  CD 4D ØØ   CALL   ØØ4D
9ØE1  21 ØD 1B   LD     HL,1BØD
9ØE4  CD 4A ØØ   CALL   ØØ4A
9ØE7  3C         INC    A
9ØE8  3C         INC    A
9ØE9  CD 4D ØØ   CALL   ØØ4D
9ØEC  C3 49 9Ø   JP     9Ø49
9ØEF  21 ØC 1B   LD     HL,1BØC
9ØF2  CD 4A ØØ   CALL   ØØ4A
9ØF5  3C         INC    A
9ØF6  3C         INC    A
9ØF7  CD 4D ØØ   CALL   ØØ4D
9ØFA  21 ØD 1B   LD     HL,1BØD
9ØFD  CD 4A ØØ   CALL   ØØ4A
91ØØ  3D         DEC    A
91Ø1  3D         DEC    A
91Ø2  CD 4D ØØ   CALL   ØØ4D
91Ø5  C3 49 9Ø   JP     9Ø49
91Ø8  21 ØC 1B   LD     HL,1BØC
91ØB  CD 4A ØØ   CALL   ØØ4A
91ØE  3D         DEC    A
91ØF  3D         DEC    A
911Ø  CD 4D ØØ   CALL   ØØ4D
9113  21 ØD 1B   LD     HL,1BØD
9116  CD 4A ØØ   CALL   ØØ4A
9119  3D         DEC    A
911A  3D         DEC    A
911B  CD 4D ØØ   CALL   ØØ4D
911E  C3 49 9Ø   JP     9Ø49
9121  CD 6C ØØ   CALL   ØØ6C
9124  C9         RET
```

Los que no dispongáis de un ensamblador, o los que no conozcáis correctamente su funcionamiento, emplead el siguiente cargador.

```
1 FOR T=&H8FFE TO &H9123
2 READ S
3 V=V+1
4 NEXT T
1ØØØ DATA 14,1,6,242,2Ø5,71,Ø,
     62,15,5Ø,233,243,62,1,5Ø,
     234
1Ø1Ø DATA 243,5Ø,235,243,2Ø5,
     98,Ø,2Ø5,114,Ø,33,Ø,56,17,Ø,
     16Ø
1Ø2Ø DATA 26,2Ø5,77,Ø,35,19,
     254,17Ø,194,3Ø,144,33,12,
     27,62
1Ø3Ø DATA 1ØØ,2Ø5,77,Ø,33,13,
     27,62,1ØØ,2Ø5,77,Ø,33,14,27
1Ø4Ø DATA 62,3,2Ø5,77,Ø,33,15,
     27,62,7,2Ø5,77,Ø,118,62,Ø,
     2Ø5
1Ø5Ø DATA 216,Ø,254,255,2Ø2,33,
     145,Ø,62,Ø,2Ø5,213,Ø,254,1
1Ø6Ø DATA 2Ø2,133,144,254,3,
     2Ø2,147,144,254,5,2Ø2,161,
     144,254
1Ø7Ø 7,2Ø2,175,144,254,2,2Ø2,
     189,144,254,4,2Ø2,214,144
1Ø8Ø DATA 254,6,2Ø2,239,144,
     254,8,2Ø2,8,145,195,73,144,
     33,12
1Ø9Ø DATA 27,2Ø5,74,Ø,61,61,
     2Ø5,77,Ø,195,73,144,33,13,
     27,2Ø5
11ØØ DATA 74,Ø,6Ø,6Ø,2Ø5,77,Ø,
     195,73,144,33,12,27,2Ø5,74
111Ø DATA Ø,6Ø,6Ø,2Ø5,77,Ø,195,
     73,144,33,13,27,2Ø5,74,Ø,61,
     61,2Ø5,77,Ø,195,73,14,4,33,
     12,27,2Ø5,74,Ø,61,61
112Ø DATA 2Ø5,77,Ø,33,13,27,
     2Ø5,74,Ø,6Ø,6Ø,2Ø5,77,Ø,
     195,73
113Ø DATA 144,33,12,27,2Ø5,74,
     Ø,6Ø,6Ø,2Ø5,77,Ø,33,13,27,
     2Ø5
114Ø DATA 74,Ø,6Ø,6Ø,2Ø5,77,Ø,
     195,73,144,33,12,27,2Ø5,74,
115Ø DATA Ø,6Ø,6Ø,2Ø5,77,Ø,33,
     13,27,2Ø5,74,Ø,61,61,2Ø5,77
116Ø DATA Ø,195,73,144,33,12,27,
     2Ø5,74,Ø,61,61,2Ø5,77,Ø,33,
     13,27,2Ø5,74,Ø,61,61,2Ø5,77,
     Ø,195,73,144,2Ø5,1Ø8,Ø
```

Teclea y ejecuta ahora el siguiente cargador que contiene los datos necesarios para la definición del sprite.

```
1Ø FOR T=&HAØØØ TO
   &HAØØØ+31
2Ø READ S
3Ø POKE T,S
4Ø NEXT T
5Ø END
6Ø DATA Ø,Ø,128,192,224,24Ø,
   255,255,112,127,Ø,Ø,Ø,Ø,Ø,Ø,
   Ø,Ø,Ø,Ø,Ø,224,252,59,2,54,Ø,Ø,
   Ø,Ø,Ø,Ø
```

Una vez cargado el programa en memoria del ordenador no olvidéis grabarlo ya sea en disco o cassette, para ello debéis teclear lo siguiente:

BSAVE "NOMBRE",&H8FFE, &HA032

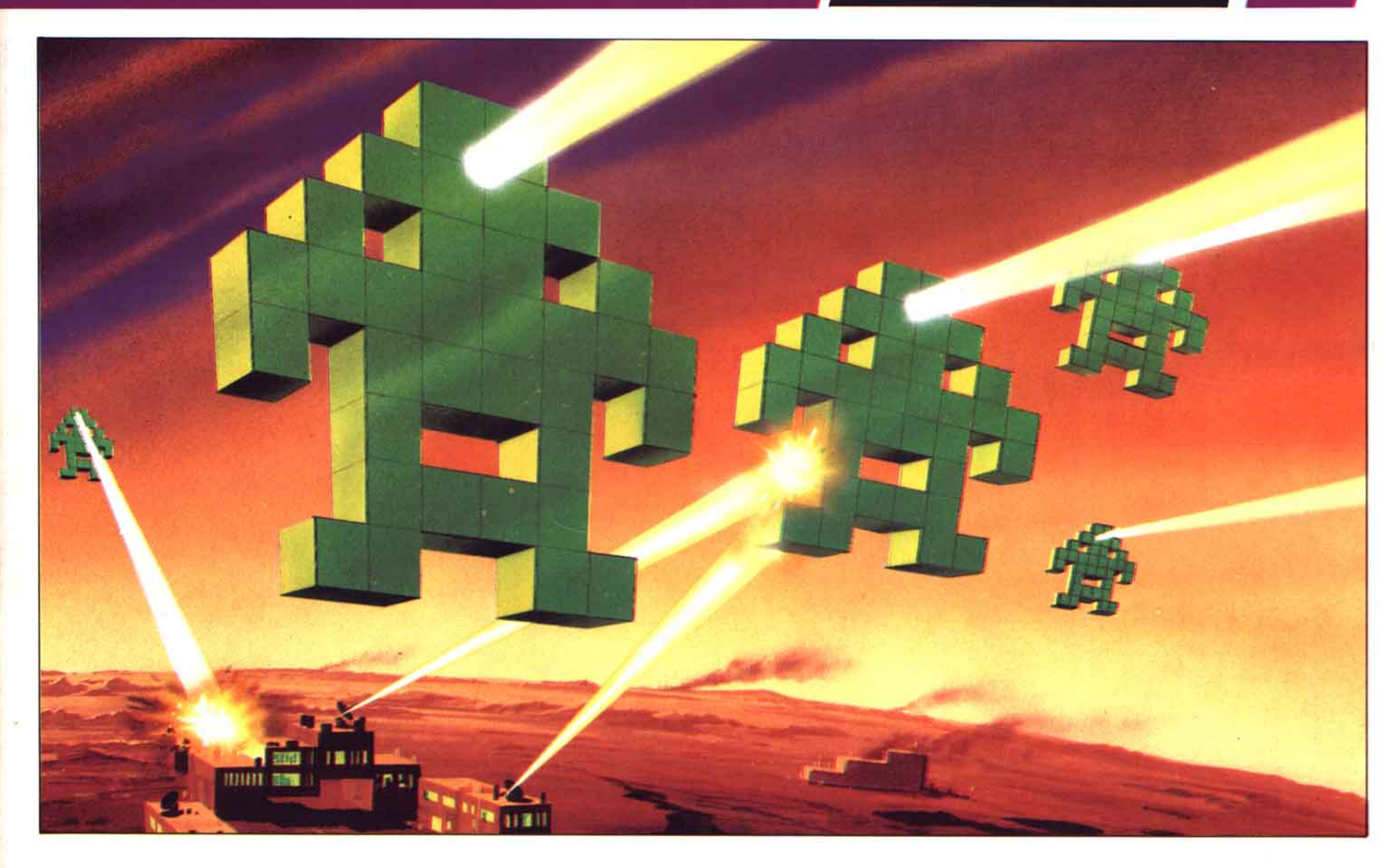
después de esto aparecerá en pantalla el conocido OK, ahora ya podemos ejecutar el programa, para ello teclea lo siguiente:

def usr =&H8FFE:X=USR(0)

Una vez hecho esto aparecerá en pantalla el dibujo de un sprite de 16×16 con forma de avión. Prueba a presionar las teclas del cursor, y verás que el sprite se desplaza por la pantalla. Para volver al BASIC sólo tienes que pulsar la barra espaciadora.

Pasemos a comentar algunas direcciones interesantes, con las que podrás alterar el resultado final del programa:

–POKE &H9045,(color deseado del sprite)
–POKE &H9049,(con contenido 0 se incrementa la velocidad, con contenido &H76 se decrementa la velocidad)
–POKE &H9007,(color deseado para los caracteres)
–POKE &H900B,(color deseado



para el fondo y el borde)

También podrás alterar la configuración del sprite alterando las direcciones de memoria comprendidas entre &HA000-&HA031.

Observando el listado en assembler vemos que abundan las llamadas a la ROM en la zona denominada BIOS. Pasemos ahora a explicar algunas de ellas:

- call 47h modifica un registro del VDP
- call 62h modifica los colores
- call 72h activa el screen 2
- call 4dh actúa como el vpoke en basic
- call d8h examina el estado del disparador para juegos
- call d5h examina el estado de las teclas de cursor
- call 4ah actúa como el vpeek en basic
- call 6ch activa el screen 0

GANADORES DE "TRUCOS DEL USUARIO DE MSX"

	NOMBRE	LOCALIDAD	JUEGO ADJUDICADO
N.º 13	Jordi Codina Claret	Manresa (Barcelona)	Penguin Adventure
	Xavier Obradors	Barcelona	Penguin Adventure
	Xavier García	Manresa (Barcelona)	Vampire Killer
N.º 15	Luis Sánchez Sabater	Castellón	Space Shuttle
	Sebastián Briones M.	Castellón	James Bond 007
	Juan Matencio P.	Barcelona	Penguin Adventure
N.º 16	Josep Ferrerons	Barcelona	Penguin Adventure
	Angel Serna	Barcelona	Némesis
	Alberto Badanas P.	Cuenca	Vampire Killer

TODO SOBRE...

HOWARD EL PATO

HOWARD THE DUCK es la nueva aventura con la que nos ha sorprendido la ya conocida casa ACTIVISION.

Esta aventura, aun siendo del tipo arcade, se escapa un poco a la trama a las que este tipo de aventuras nos tiene acostumbrados.

De hecho no nos veremos acosados por infinidad de enemigos que nos harán la vida imposible, ni tendremos que andar matando a diestro y siniestro, como si de la mejor película de guerra se tratara. Habrá enemigos, eso sí, pero con moderación. En resumidas cuentas, para llevar a buen fin este programa, sólo nos hará falta un poco de imaginación, y sobre todo mucha habilidad.

LA HISTORIA

HOWARD es lanzado en paracaídas sobre una isla desierta, bueno, casi desierta, pues escondidos en cavernas bajo tierra se encuentran los aborígenes de dicha ínsula, que nos darán más de un susto. Aterrizará aparatosamente en la parte suroeste de la isla, y su misión será llegar sano y salvo, es un decir, justamente al lado opuesto, para lo cual tendrá que sortear diferentes obstáculos, entre ellos los nativos susodichos, lagunas infestadas de alimañas y otros bichos que deberá cruzar, etcétera.

En la isla se hallan ciertas zonas por las que no podrá seguir a pie, y por ello deberá recurrir a sus potentes saltos. Dichos saltos, os lo podemos asegurar, pueden llevarnos por la calle de la amargura, ya que no son tan fáciles de ejecutar y, en el peor caso, veremos, en el colmo de nuestra desesperación, cómo nos quedamos cortos, volviendo HOWARD al punto de partida.



Pero sin duda alguna el mayor contratiempo con el que nos vamos a encontrar en esta aventura es el tiempo, restringido y fatal, que nos impedirá multitud de veces acabar la partida por su culpa.

En resumidas cuentas, éste es un buen programa de ACTIVISION, en el que nos encontramos unos gráficos bastante aceptables y un *scroll* de pantalla muy bien logrado.

CONSEJOS UTILES

Los nativos: saldrán de las cuevas que hay bajo tierra atacándonos sin piedad cada vez que presientan nuestra presencia cerca de sus guaridas. De todas formas, siempre hay un factor común en los ataques de que seamos objeto, independientemente del lugar de donde salgan los nativos: éstos casi siempre se desplazarán hasta nuestra derecha, e intentarán acabar con nosotros desde ese ángulo.

Los saltos: para efectuar dichas filigranas aeróbicas usad los cursores del ordenador y no el inseguro joystick. No hay red para los osados ni hospital para los estrellados. Para ejecutarlos pulsaremos simultáneamente las te-

clas del cursor de abajo y derecha o izquierda, dependiendo esto de la dirección que le queramos dar al salto. A la vez que hacemos esto también tendremos que pulsar el botón de disparo (o la barra espaciadora). A pesar de esto observaremos que a veces el salto no se produce; ello es debido a que sólo hay una posición en el que se efectuará.

El agua: para atravesar las lagunas nos hará falta un objeto que se halla en la zona derecha a partir del lugar en el que aterrizamos, mejor dicho, en el que nos golpeamos con el paraguas que llevamos por paracaídas. Recomendamos que esto sea lo primero que se haga al comenzar el juego.

Matar aborígenes: podemos eliminar a los nativos (colonialismo pesticida) situándonos de cara a la dirección por donde nos atacan y pulsando la barra espaciadora (o el *trigger*) y los cursores de derecha o izquierda repetidas veces, dependiendo esto último del lugar desde donde nos ataquen; generalmente suele ser por la derecha.

Por último, os daremos un último y valioso consejo: en la zona final de la isla, justo cuando nos disponemos a cruzar un puente, empezarán a caer ciertos objetos; recomendamos fervorosamente esquivarlos, de lo contrario regresaremos a la zona anterior. Cuando consigamos pasar dicho obstáculo nos encontraremos con un agujero en el suelo idéntico al de los que usan los nativos por vivienda, sólo que en este caso en dicho socavón no hay nadie. Nos acercaremos a él cuidadosamente, y dando un pequeño salto lograremos introducirnos en él, con lo que el juego habrá terminado.



EL CARGADOR

```
 1Ø REM----INPUT MSX----
 2Ø REM----POR------
 3Ø REM----JUANMA-----
 4Ø CLS:KEYOFF
 5Ø FORI=&H85ØØTO&H8512
 6Ø READ A$
 7Ø A=VAL("&H"+A$)
 8Ø POKE I,A
 9Ø NEXT
1ØØ FOR I=&H855Ø TO&H855D
11Ø READ A$
12Ø A=VAL("&H"+A$)
13Ø POKE I,A
14Ø NEXT
15Ø LOCATE4,1Ø:INPUT"QUIE-
    RES TIEMPO INFINITO";A$
16Ø IF A$="S" OR A$="s" THEN
    17Ø ELSE22Ø
17Ø CLS
18Ø POKE&H85Ø1,Ø:
    POKE&H85Ø9,Ø
19Ø BLOAD"cas:",&HØØØ
2ØØ POKE&HD422,Ø:
    POKE&HD423,&H85
21Ø DEFUSR=&H855Ø:A=USR(Ø)
22Ø BLOAD"cas:",R
23Ø DATA 3E,D6,32,D2,C9,32,DD,
    C9,3E,1,32,D3,C9,32,DE,C9,
    C3,C8,AF
24Ø DATA 21,Ø,DØ,11,Ø,8Ø,1,24,
    4,ED,BØ,C3,Ø,8Ø
```





SERVICIO DE EJEMPLARES ATRASADOS



N. 2 Mi ordenador ya sabe dibujar. Enviando mensajes secretos. Juegos de laberinto. Traductor de texto. Almacenamiento de programas en BASIC. Rutinas de tanteo y tiempo. Software de allende las fronteras.



N. 6 La memoria de video. Todo sobre READ y DATA. El lenguaje de la tortuga. Puzzles y matemáticas. Estructura tus programas. Taller de hardware. Inteligencia artificial en tu micro.



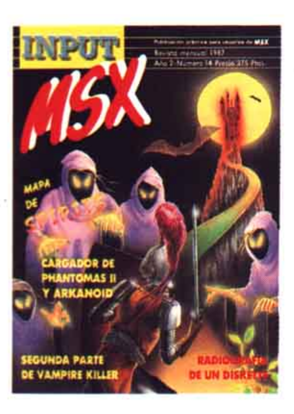
N. 10 Lectura del directorio de un disco. Arquitectura de la Unidad Central de Procesos (CPU). Código máquina para todos. Generador de discursos. El ordenador y el teléfono. LISP e inteligencia artificial.



N. 11 Dibujando bit a bit. Entendiendo el PEEK y el POKE. Creador automático de menús. Mapa y pokes para LIVINGSTONE SUPONGO. SWAPS: el juego de los números invertidos. Biblioteca de datos.



N. 12 Diseñador de teclado. Lector de cabeceras de cinta. Música, micros y midi. Mapa y pokes para FUTURE KNIGHT. Examina la memoria de tu micro. Las serpientes sumadoras.



N. 14 Radiografía de un diskette. Las variables del sistema. Bueno como el oro. Mapa de SPIRITS. Ultimas novedades del software internacional. Arquitectura interna del ordenador.

¡NO TE PIERDAS NI UN SOLO EJEMPLAR!

INPUT MSX quiere proporcionar a sus lectores este servicio de números atrasados para que no pierdan la oportunidad de tener en sus hogares todos los ejemplares de esta revista, líder en el mercado español. Podréis solicitar cualquier número de INPUT MSX que deseéis, siempre al precio de cubierta (sin más gastos).

Utiliza el cupón adjunto, enviándolo a EDISA (Dpto. de Suscripciones), López de Hoyos, 141 -28002 MADRID.

CUPON DE PEDIDO

Sí, envíenme contra reembolso......... ejemplares de INPUT MSX de los números:

(escriba en letra de imprenta)

....

NOMBRE

APELLIDOS

DOMICILIO

NUM. PISO COD. POSTAL

POBLACION PROV.

TELEFONO FIRMA

JUGANDO A LA GUERRA (y II)

- MAPAS, HOMBRES, ORDENES Y BATALLAS
- DESARROLLO DE LA ACCION
- FACTORES QUE AFECTAN AL MOVIMIENTO DE LAS TROPAS

En el capítulo anterior se establecieron los símbolos de cada una de las unidades militares necesarias para el combate. Éstos se representan (y mueven) sobre el mapa para indicarnos el progreso y estrategia del juego.

El programa posee unas variables para representar el mapa. Como éste se visualiza durante todo el tiempo, dichas variables son innecesarias. Siempre habrá un área de memoria reteniendo la pantalla con toda la información.

Las variables tienen tantos elementos como posiciones de pantalla hay dentro del mapa. Una variable normal necesita unos cinco bytes por elemento, y una entera, cuatro bytes.

LAS TROPAS

Para identificar la situación de las unidades en el mapa (para que puedan ser movidas y detenidas si hay obstáculos en el camino), se necesitará otra variable: la variable de tropa. Dicha variable contiene la información que se precise para el combate, estado de ánimo, movimiento, etc.

FIJANDO LAS VARIABLES

Las siguientes rutinas dimensionan las variables correspondientes al mapa y a la tropa:

```
35Ø  DIM M(16,27)
355  DIM T(16,9)
```

RELLENANDO EL MAPA

El siguiente paso consiste en determinar el terreno y las posiciones de partida de cada unidad y mostrarlas en pantalla.

El modo más sencillo para ello es utilizar el generador de números aleatorios del ordenador.

ESCOGIENDO EL TERRENO

La rutina de **elección de terreno** es esencialmente aleatoria, pero existe un grado de control sobre la elección. Esto asegura que el terreno sea trazado de forma realista con bosquecillos, colinas, ríos, etc.

```
17  CG=RND (-TIME): CG=(CG*
    1Ø)
18  FOR FF=1 TO CG
2Ø  DEF FN R(X)=INT(RND(1)*
    X)+1
21  NEXT
```

BAJO CONTROL

Veamos la rutina de elección de terreno en la que se va a desarrollar la batalla.

```
37Ø  I$="nose"
47Ø  REM ** CREANDO **
475  FOR XF=1 TO 15
476  LOCATE 1,XF:
     PRINT"FFFFFFFFFFFF
     FFFFFFFFFFFFFFF"
477  NEXT
48Ø  FOR I=1 TO 16: GOSUB 8ØØ:
     M(I,1)=R: NEXT I
49Ø  FOR I=1 TO 15
5ØØ  FOR J=2 TO 27
51Ø  S=FN R(1Ø)
52Ø  IF S<8 THEN GOSUB 8ØØ
525  IF S>=8 THEN R=M(I,J-1)
53Ø  M(I,J)=R
54Ø  IF R=3 AND J<27 THEN
     M(I,J+1)=4
55Ø  IF M(I,J)<>Ø AND M(I,J)<>3
     THEN LOCATE J,I:
     PRINTCHR$((M(I,J)+64))
555  IF M(I,J)=3 AND J<>27 THEN
     LOCATE J,I: PRINTCHR$(67):
     LOCATE J+1,I:
     PRINTCHR$(68)
56Ø  NEXT J
57Ø  NEXT I
585  GOSUB  72Ø
59Ø  FOR I=1 TO 8
6ØØ  FOR J=1 TO 2
6Ø5  T(I+8,J)=2: NEXT J
61Ø  FOR J=3 TO 4: READ T (I,J)
62Ø  READ MR
63Ø  FOR J=Ø TO 8 STEP 8
64Ø  T(I+J,5)=MR+FNR (2)
65Ø  T(I+J,6)=(FN R(1ØØ)*1Ø)+1Ø
66Ø  T(I+J,7)=T(I+J,6)
67Ø  NEXT J
68Ø  T(I,8)=15
69Ø  T(I+8,8)=1
7ØØ  NEXT I
7Ø5  IF TT1=2 THEN GOSUB 29ØØ
7Ø6  IF TT1=3 THEN GOSUB 291Ø
7Ø7  IF TT1=4 THEN GOSUB 292Ø
71Ø  RETURN
```

DESARROLLO DE LAS TROPAS

Las posiciones de las tropas están contenidas en la variable de tropa como un par de coordenadas: horizontal y vertical.

Las posiciones de partida de las unidades oponentes se hallan en diferentes extremos del mapa: la posición inicial del jugador en el extremo sur, y la del ordenador, en el extremo norte.

```
86Ø  REM  ** DISPONER LAS
     TROPAS **
87Ø  REM
88Ø  FOR M=1 TO 2
89Ø  S=1: R=1
9ØØ FOR K=1 TO 8
91Ø  REM
92Ø  S=FNR(8*M)
93Ø  IF T(S,9)<>Ø  THEN GOTO
     91Ø
94Ø  R=FN R(4)+R
95Ø  R=R- INT(R/27)
96Ø  T(S,9) = R
97Ø  REM
98Ø  LOCATE T(S,9),T(S,8)
985  PRINTMID$ (U$,S,1)
99Ø  NEXT K
1ØØØ  NEXT M
1Ø1Ø  RETURN
```

SOBRE EL CAMPO DE BATALLA

Una vez que han sido determinadas las posiciones de partida de ambos ejércitos, las unidades están preparadas para ser mostradas en pantalla.

```
410 U$=CHR$(72)+CHR$(73)+
    CHR$(74)+CHR$(74)+
    CHR$(75)+CHR$(75)+
    CHR$(76)+CHR$(76)
411 U1$=CHR$(80)+CHR$(81)+
    CHR$(82)+CHR$(82)+
    CHR$(83)+CHR$(83)+
    CHR$(84)+CHR$(84)
412 U$=U$+U1$
413 VPOKE8201,143:VPOKE
    8196,16:VPOKE8218,240
414 VPOKE8202,79:VPOKE
    8217,240
```

EL RECINTO HOSTIL

La pantalla puede volverse más atractiva si se dibuja una frontera limitando el campo de batalla.

```
720 REM **DECORACION DEL
    BORDE **
730 FOR I=0 TO 16
740 LOCATE 0,I: PRINT CHR$(69)
741 LOCATE 28,16-I:PRINT CHR$(69)
750 NEXT I
760 FOR I=0 TO 28
770 LOCATE I,0: PRINT CHR$(69)
771 LOCATE 28-I,16: PRINT
    CHR$(69)
780 NEXT I
790 RETURN
```

MOVILIZANDO LAS FUERZAS

Ahora que el mapa está trazado, ambas partes desearían mover sus ejércitos. Estos se desplazan respondiendo a otras órdenes, pero hay diversos presupuestos que deben ser examinados antes de que una unidad pueda ser movida dentro del campo de batalla.

El programa debe saber cuál es el máximo movimiento (número de cuadrados) para cada unidad. La movilidad de éstas depende únicamente del peso del armamento, pero también podrían influir en ella la disciplina, la moral, el cansancio...

Con la siguiente rutina el programa podrá comprobar dichos factores antes de poner en movimiento alguna unidad.

```
1160 REM **MOVIMIENTO DE LA
     UNIDAD **
1170 OX=T(B,8): OY=T(B,9)
1175 Z$=" "
1180 IF M(T(B,8),T(B,9))<>0
     THEN Z$=CHR$(64+M
     (T(B,8),T(B,9)))
1190 D=5- T(B,4)
1200 IF B<3 OR B=9 OR B=10
     THEN D=D+2
1210 V=T(B,2)-1
1215 UP=0: AL=V-2
1220 IF (V/2-(INT(V/2)))=0 THEN
     UP=V-1: AL=0
1230 REM
1240 N1=T(B,9)+AL:
     NP=T(B,8)+UP
1250 IF NP<1 THEN NP=1
1260 IF NP>15 THEN NP=15
1270 IF N1<1 THEN N1=1
1280 IF N1>27 THEN N1=27
1290 IF M(NP,N1) > 0 THEN D=D-1
1300 FOR K=1 TO 8
1310 IF (T(K,9)=N1 AND
     T(K,8)=NP ANDK<>B)
     THEN D=0
1315 IF (T(K+8,9)=N1 AND
     T(K+8,8)=NP AND
     K+8<>B) THEN D=0
1320 NEXT K
1330 IF D>0 THEN T(B,9)=N1:
     T(B,8)=NP: D=D-1
1340 IF D<>0 THEN GOTO 1230
1350 LOCATE OY,OX: PRINTZ$
1355 IF Z$=" " THEN LOCATE
     OY,OX;PRINT"F"
1360 LOCATE T(B,9),T(B,8):
     PRINTMID$(U$,I,1)
1370 RETURN
```

EL ARTE DE COMANDAR

Hay un número de factores conectados con las tropas que afectarán a la manera en que éstas actúen en el campo de batalla. Veamos algunos de ellos:

–Orden actual de la unidad (lo último que se les ordenó que hicieran);
–Dirección del movimiento actual;
–Armamento y armaduras;
–Fuerza: inicial y actual;
–Moral o actitud;
–Posición y terreno.

Estos factores corresponden a los elementos de la variable de tropa que establecimos anteriormente.

VISITANDO EL CUARTEL GENERAL

Esta rutina inicializa los elementos no completados de la variable:

```
190 REM ** INICIALIZACION **
200 VC=0: DE=0
310 REM ** definición de matrices
    **
320 COLOR ,15,6
330 COLOR 1
340 CLS
360 DIM T$(8,12): DIM O$(5,12):
    DIM W$(5,9)
361 DIM M$(5,12): DIM A$(4,12):
    DIM R$(4,12):  DIM C(8)
385 READ O1$: READ W1$: READ
    M1$
386 O$(I,12)=O1$:  W$(I,9)=W1$:
    M$(I,12)=M1$
387 NEXT I
389 RESTORE2790
390 FOR I=1 TO 8
395 READ T1$: T$(I,12)=T1$
396 NEXTI
399 RESTORE2800
400 FOR I=1 TO 4
405 READ A1$: READ R1$
406 A$(I,12)=A1$:  R$(I,12)=R1$
407 NEXT I
415 X$="NnSs"
420 RETURN
2760 DATA "fuego","ningun",
     "cobarde","alto","arco",
     "indispuesta"
2770 DATA "movimiento",
     "espada","dispuesta",
     "estado","hacha","brava"
2780 DATA "derrota","lanza",
     "valiente"
2790 DATA "caballeros",
     "sargentos","lanceros",
     "lanceros","arqueros",
     "arqueros","aldeanos",
     "aldeanos""
2800 DATA "ningun","llanuras",
     "chaquet"n","pueblo",
     "cotaFmalla","bosque",
     "mesetas","colinas"
2810 DATA 5,4,3,5,3,3,4,3,2,3,3,1,
     2,2,1,2,3,2,3,2,0,3,1,0
```

DANDO ORDENES

La totalidad del juego depende de las órdenes dadas por el jugador al ejército; sin éstas no habría combate y, por consiguiente, ni vencedores ni vencidos.

Hay cuatro órdenes que podemos dar a las tropas: fuego, hacer alto, movimiento y status. La orden de abrir fuego sólo se aplica a los arqueros, y la cumplirán aun cuando no exista un blanco en los alrededores. Si seleccionamos abrir fuego, y la unidad no es de arqueros, entonces se nos solicitará otra orden; la orden de hacer alto es explícita por sí misma; la orden de movimiento solicita la entrada de una determinada dirección; en lo referente al status, éste responde absolutamente a la descripción del estado actual de la unidad.

SELECCIONANDO UNA UNIDAD

Cuando es el turno del jugador, la primera unidad se ilumina, seguida en turno por las otras siete.

```
1380 REM ** Selecciona la unidad
     para órdenes **
1390 GOSUB 2540
1400 REM
1401 LOCATE 7,21: PRINT"
```



```
      pulsaFunaFtecla"
1403 LOCATE T(I,9),T(I,8):
      PRINT"F"
1405 FOR TL=1 TO 50:NEXT TL
1410 LOCATE T(I,9),T(I,8):
      PRINTMID$(U$,I,1)
1411 FOR TJ=1 TO 50:NEXT TJ
1415 LET G1$=INKEY$: IF G1$=""
      THEN  GOTO 1403
1417 GOSUB 2540
1418 FOR LP=18 TO 20: LOCATE
      0,LP
1419 PRINT"FFFFFFFFFFFFFFFFF
      FFFFFFFFFFF": NEXT LP
1420 LOCATE 0,18: PRINT"unidad
      FnumeroF"+STR$(I)+"F"+T$
      (I,12)+"FFFF"
1425 LOCATE 14,18: PRINT"F"
1430 LOCATE 0,19:
      PRINT"lasFordenesFson";
      "F";O$(T(I,1),12)
1440 IF T(I,1)=3 YHEN LOCATE
      28,19: PRINTMID$(I$,T(I,2),1)
1450 REM ** Variables para el
      ciclo **
1460 LOCATE 0,20: PRINT
      "cambiaFlasFordenesFF(s/n)
      ?"
1465 Y=0: Y$=INKEY$
1466 IF Y$="" THEN GOTO 1465
1470 FOR K=1 TO 4
1475 IF MID$(X$,K,1)=Y$ THEN
      Y=K
1480 NEXT K
1485 IF X$(K)=Y$ THEN Y=K
1490 IF Y=0 THEN GOTO 1450
1500 RETURN
```

EMITIENDO ORDENES

Esta rutina muestra las órdenes opcionales.

```
1900 REM ** Selecciona   accion
      **
1910 GOSUB 2540
1920 LOCATE 1,18:
      PRINT"opcionesFson:"
1930 FOR J=1 TO 4
1935 LOCATE 15,17+J:
      PRINT"FFFFFFFFFFFFF"
1940 LOCATE 15,17+J:
      PRINTLEFT$(O$(J,12),1);
      "-";O$(J,12)
1950 NEXT J
1960 REM ** Simulacro de bucle
      **
1962 A=0
1965 F$="FfAaMmEe"
1970 G$=INKEY$: IFG$="" THEN
      GOTO 1970
1975 FOR K=1 TO 8
1980 IF MID$(F$,K,1)=G$ THEN
      A=INT((K+1)/2)
1985 NEXTK
1990 IF A<=0 THEN GOTO 1960
2000 IF I<>6 AND I<>5 AND
      A=1  THEN GOSUB 2540:
      LOCATE 10,18:
      PRINT"ningunFarco":
      GOSUB 2410: GOTO 1910
2010 IF A=4 THEN GOSUB 2440:
      RETURN
2020 T(I,1)=A
2030 IF A=3 THEN GOSUB 2050
2040 RETURN
```

UNA NUEVA DIRECCION

Si el jugador emite una orden de movimiento, deberá asignársele también a ésta una dirección. Esta rutina maneja las opciones de movimiento.

```
2050 REM **Decide la dirección
      del movimiento **
2055 GOSUB 2540
2060 LOCATE 0,19: PRINT"
      queFdirecci"n:"
2061 LOCATE 20,18: PRINT"n"
2062 LOCATE 20,17:
      PRINTCHR$(206)
2063 LOCATE 18,19:
      PRINTCHR$(208)+"o "
2064 LOCATE 21,19:  PRINT"e"+
      CHR$(207)
2065 LOCATE 20,20: PRINT"s"
2066 LOCATE 20,21:
      PRINTCHR$(205)
2068 G=0
2070 REM ** Simulacro para
      bucle **
2080 G$=INKEY$: IF G$="" THEN
      GOTO 2080
2090 IF ASC(G$)<96 THEN
      G$=CHR$(ASC(G$)+32)
2095 FOR K=1 TO 4
2100 IF MIF$(I$,K,1)=G$ THEN
      G=K
2105 NEXT K
2110 IF G=0 THEN GOTO
      2070
2120 T(I,2)=G
2130 RETURN
```

STATUS: ESTADO

Podemos saber el estado de cualquier unidad, mientras planeamos el gran ataque, seleccionando la opción status.

Serán mostrados todos los elementos de la variable de tropa (o sus equivalentes en palabras).

```
2440 REM ** ESTADO DE LA
      UNIDAD **
2450 GOSUB 2540
2460 LOCATE 0,17:
      PRINT"unid.:"+STR$(I)
      +"FFFFtipo.:"+T$(I,12)+
      "FFFFFFF"
2465 LOCATE 6,17: PRINT"F"
2470 LOCATE 0,18: PRINT"arma.:
      "+W$(T(I,3),9)+
      "FFFFFFFFFF"
2480 LOCATE 12,18: PRINT"prtc.:
      "+A$(T(I,4),12)+
      "FFFFFFFFFF"
2490 LOCATE 0,19: PRINT"frza.:
      "+STR$(T(I,7))+
      "FFFFFFFFF"
2495 LOCATE 6,19: PRINT"F"
2500 LOCATE 12,19: PRINT"acti.:
      "+M$(T(I,5),12)+
      "FFFFFFFFF"
2510 LOCATE 0,20: PRINT"locz.:
      "+R$((M(T(1,8),T(I,9))+1),12)
      +"FFFFFFFFFFFFFFF"
2520 GOSUB 2410
2530 GOTO 1900
```

EL EFECTO DE LAS ORDENES

La rutina siguiente comunica al jugador si una determinada unidad está cumpliendo la orden encomendada.

```
1020 REM **EFECTO DE LAS
      ORDENES **
1030 FOR I=1 TO 16
1032 IF T(I,1)>3 THEN GOTO 1140
1035 GOSUB 2540: LOCATE 2,19
1037 PRINT"unidadF"+STR$(I)
      +"FdecidesFactuar"
1038 LOCATE 9,19: PRINT"F"
1040 CL=1: IF I>8 THEN CL=2
1050 IF T(I,1)=3 THEN B=I:
      GOSUB 1160
1055 IF T(I,1)=2 THEN GOTO 1140
1060 IF T(I,1)=1 THEN SH=I:
      GOSUB 1710: GOTO 1140
```

```
1Ø7Ø  FOR F=-1 TO 1
1Ø8Ø  FOR G=-1 TO 1
1Ø9Ø  FOR E= 1 TO 16
11ØØ  IF (T(I,8)+F=T(E,8))   AND
      (T(I,9)+G=T(E,9)) AND
      T(E,1)<>5 THEN US=I:
      TH=E: GOSUB 151Ø
111Ø  NEXT E
112Ø  NEXT G
113Ø  NEXT F
114Ø  NEXT I
115Ø  RETURN
```

EL ORDENADOR COMO OPONENTE

El ordenador da las órdenes a sus unidades de un modo más o menos arbitrario.

```
214Ø  REM** EL ENEMIGO
      SELECCIONA LA ACCION **
215Ø  T(E,2)=3
216Ø  T(E,1)=FN R(3)
217Ø  IF T(E,1)=1 AND T(E,3)<>2
      AND  T(E,1)<>5 THEN
      GOTO 216Ø
218Ø  IF T(E,1)=3 AND T(E,1)<>5
      THEN IF FN R(2)=1 THEN
      T(E,2)=FN R(4)
219Ø  RETURN
```

PROYECTILES

Existen dos clases diferentes de combate: proyectiles (flechas) y cuerpo a cuerpo. Esta primera rutina trata el combate con proyectiles.

```
171Ø  REM ** RUTINA DE
      PROYECTILES **
172Ø  GOSUB 254Ø
173Ø  LOCATE 4,18:
      PRINT"laFunidadF"+
      STR$(SH)+F"disparo"
1735  LOCATE 14,18: PRINT"F"
174Ø  FX=5:
      FY=5: GP = -1
1745  ST=9
175Ø  IF SH>8
      THEN ST=1
177Ø  FOR M=ST
      TO (ST+7)
178Ø  TM=ABS
      (T(M,8)-T(SH,8)):
      TY=ABS
      (T(M,9)-T(SH,9))
1785  IF TM<FX AND T(M,1)
      <5 AND TY<FY THEN
      FX=TM:
      FY=TY: GP=M
179Ø  NEXT M
18ØØ  IF GP=-1
      THEN LOCATE 7,19: PRINT
      "nadaFalFalcance":
      GOSUB 241Ø:  RETURN
181Ø  C=8-T
      (GP,4)-ABS(FX-FY)
182Ø  IF GP<3
      OR GP=9 OR GP=10  THEN
      C=C+1
183Ø  IF M(T(GP,8),
      T(GP,9))=2 THEN
      C=C-2
184Ø  IF T(GP,1)
      <>2 THEN
      C=C+1
185Ø  C=
      (C+(INT(T(SH,7)/4Ø))
      +FN  R(3))*10
186Ø  T(GP,7)=
      T(GP,7)-C
1868  LOCATE Ø,
      2Ø: PRINT"FFFFFF
      FFFFFFFFFFFFFFFFFFFFFFF"
1869  LOCATE Ø,19:
      PRINT"FFFFFFFFFFFF
      FFFFFFFFFFFFFFFFF"
187Ø  LOCATE Ø,19:
      PRINT"seFhanFproducido:"
      +STR$(C)+
      "FbajasFFFFFFFF"
1871  LOCATE 17,19:
      PRINT"F"
1872  LOCATE   Ø,2Ø:
      PRINT"enFlaFunidad:"+
      STR$(GP)
1873  LOCATE 13,2Ø:
      PRINT"F"
1875  GOSUB 241Ø
188Ø  UN=GP:
      UM=GP: GOSUB 22ØØ
189Ø  RETURN
```

EL CHOQUE

El combate cuerpo a cuerpo se calcula de una forma similar.

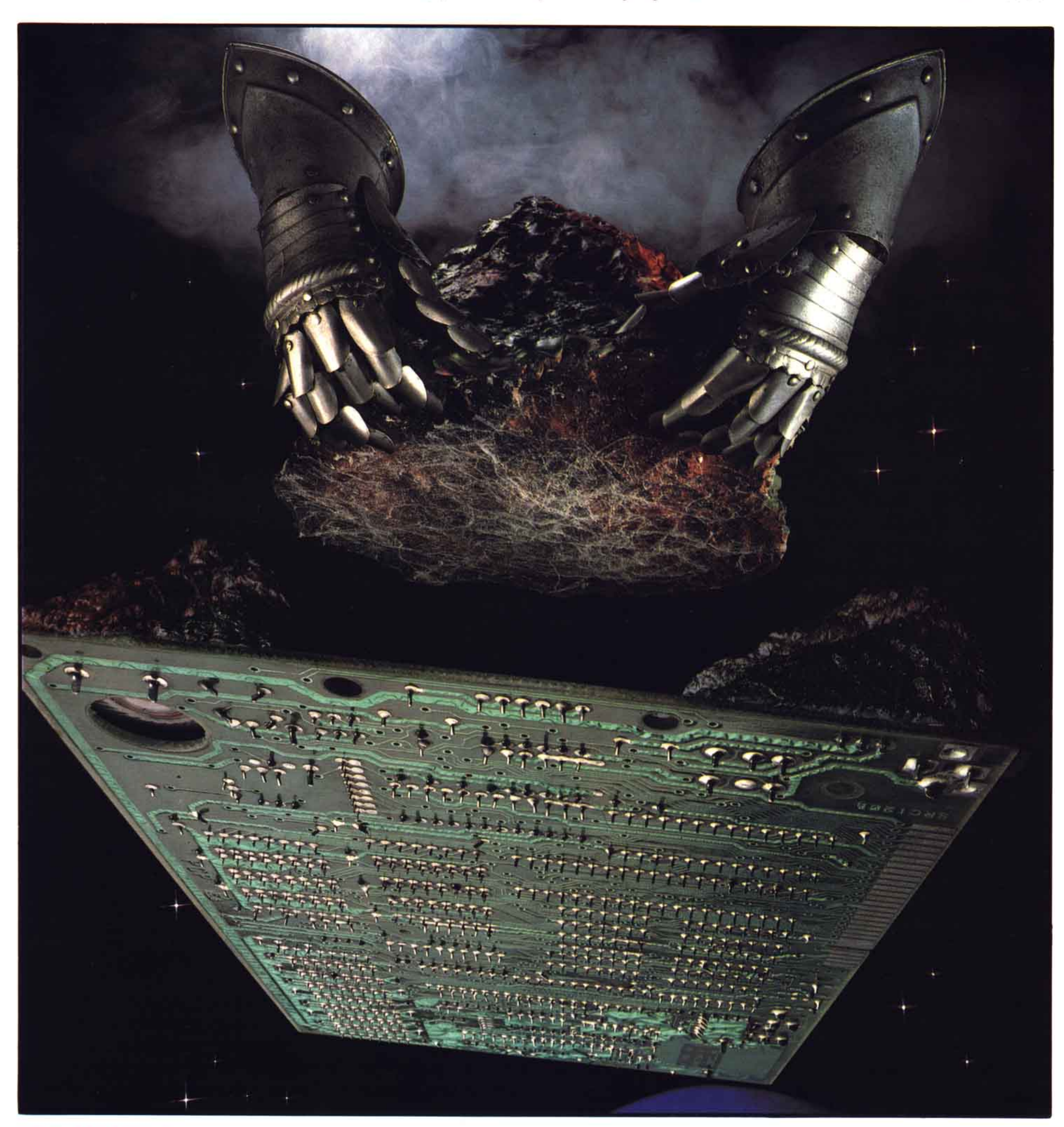
```
151Ø  REM ** COMBATE **
152Ø  IF (US<9 AND TH<9) OR
      (US>8 AND TH>8) THEN
      RETURN
153Ø  IF T(US,1)=5 OR T (TH,1)=5
      THEN RETURN
154Ø  GOSUB 254Ø
155Ø  LOCATE 8,18
1551  PRINT"——combate——"
156Ø  AT=INT((T(US,7)-T
      (TH,7))/50)
157Ø  AT=AT+T(US,3)-T(TH,4)+
      T(US,5)+FN R(5)
158Ø  IF ABS(T(US,2)-T(TH,2))<>
      2 THEN AT=AT+2
159Ø  IF US<3 OR US = 9 OR
      US=1Ø THEN AT=AT+1
16ØØ  DR=INT((T(TH,7)-T(US,7))
      /60)
161Ø  DR=DR+T(TH,3)-T(US,4)-
      T(TH,5)+M(T(TH,8),T(TH,9))
      +FN R(3)+2
1615  WN=TH: LO=US
162Ø  IF AT>DR THEN WN=US:
      LO=TH
163Ø  WC=INT(T(WN,7)/1Ø)
1631  IF WC<1 THEN WC=1
164Ø  T(WN,7)=T(WN,7)-WC
165Ø  LC=INT(T(LO,7)/5)
1651  IF LC<1 THEN LC=1
166Ø  T(LO,7)=T(LO,7)-LC
1665  FOR FK=19 TO 2Ø: LOCATE
      Ø, FK
1667  PRINT"FFFFFFFFFFFFFFFFFF
      FFFFFFFFFF":NEXT FK
167Ø  LOCATE Ø,19:
      PRINT"unidad:"+STR$(WN):
      LOCATE 15,19: PRINT
      "perdidas:"+STR$(WC)
1672  LOCATE 7,19: PRINT"F":
      LOCATE 24,19: PRINT"F"
1675  LOCATE Ø,2Ø: PRINT"
      unidad:"+STR$(LO):
      LOCATE 15,2Ø: PRINT
      "perdidas:"+STR$(LC)
1677  LOCATE 7,2Ø: PRINT"F":
      LOCATE 24,2Ø: PRINT"F"
168Ø  GOSUB 241Ø
169Ø UN=LO: UM=WN: GOSUB
      22ØØ
17ØØ  RETURN
```

La principal diferencia entre el combate cuerpo a cuerpo y el combate de proyectiles reside en que ambas partes tienen algo que decir en la afrenta. Esto significa que el combate cuerpo a cuerpo es un asunto más desordenado, que requiere dos juegos de bajas para ser calculado. A continuación, la diferencia entre las armas de los atacantes y el armamento de los

defensores es agregada, junto con el valor de la moral de los atacantes y un número aleatorio superior a cinco.

Los atacantes también ganan una bonificación si el enemigo no se enfrenta directamente a ellos. Este será el caso, tanto si la unidad no está moviéndose directamente hacia los atacantes, como –cuando la unidad ha hecho alto– si no estaba moviéndose en aquella dirección en la que se movía la última vez. Ello se debe a que el elemento de dirección de la variable de tropa nunca es eliminado, y siempre debe poseer una dirección. Históricamente, parece ser que el ataque por la retaguardia o por los flancos ha sido siempre uno de los factores más significativos del combate. Finalmente, los atacantes obtienen una bonificación de un punto si pertenecen a la caballería.

Los defensores son tratados similarmente. El factor arbitrario trata de de-



terminar el hecho de que es más fácil defender que atacar, pero algunas veces el ansia de sangre de los atacantes puede intervenir para predominar sobre la ventaja innata del hecho de defender.

TEST MORAL DE LA UNIDAD

El factor psicológico es muy importante en la guerra. Es bastante improbable que un ejército gane una batalla, aun equipado con los más poderosos carros de combate del mundo, si las tropas odian a sus generales, si creen que el enemigo tiene una causa justa, o si les repugna la idea de combatir, sea como sea. Existen miles de factores involucrados en la psicología individual de los soldados, y ningún juego ha llegado nunca a representarlos todos en su descarnada realidad. En el nivel más ridículo, la actitud de un guerrero en una pugna corriente puede depender de si ha dormido bien o no la noche anterior, o de la calidad de su última comida.

La moral no sólo puede ser complicada, sino que puede influir justamente sobre cada uno de los aspectos de la batalla, dando un empuje o bloqueando a los bandos. La moral sólo pasa el test cuando una unidad pierde un combate o es arrasada.

```
22ØØ REM ** TEST MORAL DE LA
     UNIDAD **
22Ø5 GOSUB 254Ø
22Ø6 LOCATE 1,18:
     PRINT"unidad:
     F"+STR$(UN),"Fhombres:F"
     +STR$(T(UN,7))
22Ø7 LOCATE 1,19: PRINT"unidad
     :F"+STR$(UM),"Fhombres:
     F"+STR$(T(UM,7))
221Ø GOSUB 241Ø
2215 IF T(UN,7)>1 AND
     T(UM,7)>1 THEN GOSUB
     254Ø: RETURN
222Ø GOSUB 254Ø
223Ø LOCATE 1,18: PRINT"perdidas
     FdemasiadoFgrandes"
224Ø LOCATE 1,19: PRINT"
     unidad:"+STR$(UN)+
     "Fdesintegrada"
2245 LOCATE 8,19: PRINT"F"
225Ø GOSUB 241Ø
226Ø IF T(UN,7)<=1 THEN T
     (UN,1)=5:GOTO 227Ø
2265 IF T(UM,7)<=1 THEN
     T(UM,1)=5:GOTO 2275
227Ø LOCATE T(UN,9),T(UN,8):
     PRINT"F"
2272 RETURN
2275 LOCATE T(UM,9),T(UM,8):
     PRINT"F"
228Ø RETURN
229Ø REM ** Test para victoria **
23ØØ GD=Ø: BD=Ø
231Ø FOR M=1 TO 8
232Ø IF T(M,1)<>5 THEN
     GD=GD=1
233Ø IF T(M+8,1)<>5 THEN
     BD=BD+1
234Ø NEXT M
235Ø IF BD<1 AND GD>Ø THEN
     VC=1
236Ø IF GD<1 AND BD>Ø THEN
     DE=1
237Ø RETURN
```

EL ARMISTICIO

Hay que añadir unos pocos toques finales al programa: en primer lugar una condición de victoria.

```
238Ø REM **Fin del mensaje **
239Ø IF VC=1 THEN GOSUB 295Ø:
     GOSUB 293Ø
2395 IF DE=1 THEN GOSUB 295Ø:
     GOSUB 294Ø
24ØØ RETURN
241Ø REM ** RETRASO **
242Ø LOCATE 7,21: PRINT
     "pulsaFunaFtecla"
2425 G$=INKEY$: IF G$="" THEN
     GOTO 2425
243Ø RETURN
293Ø CLS: FOR FQ=1 TO 16
2931 PRINT" ——victoria——"
2935 NEXT: RETURN
294Ø CLS: FOR FQ=1 TO 16
2941 PRINT"unFenfrentamientoF
     frustrado"
2945 NEXT: RETURN
295Ø FOR F=1 TO 23
2951 PRINT"
2952 NEXT: RETURN
```

Ahora ya tienes todas las rutinas que completan el *wargaming*. Todo lo que necesitas para gobernar a tus tropas es el bucle principal.

```
1Ø CLEAR 6ØØ
12 COLOR 15,1,2
15 SCREEN1:GOSUB 3ØØØ
3Ø GOSUB 19Ø
35 GOSUB 32ØØ
4Ø GOSUB 47Ø
5Ø GOSUB 86Ø
6Ø REM ** SIMULACRO PARA
   REPETICION DE BUCLE **
7Ø FOR I=1 TO 8
8Ø IF T(I,1)<4 AND T(I,1)<>5
   THEN GOSUB 138Ø
85 IF Y>2 AND T(I,1)<>5 THEN
   GOSUB 19ØØ
9Ø IF T(I,1)<5 AND T(I,1)<>5
   THEN LOCATE T(I,9),T(I,8):
   PRINT MID$(U$,I,1)
1ØØ NEXT I
11Ø FOR E=9 TO 16
12Ø IF T(E,1)<4 THEN GOSUB
    214Ø
13Ø NEXT E
14Ø GOSUB 1Ø2Ø
15Ø GOSUB 229Ø
16Ø IF VC<>1 AND DE<>1 THEN
    GOTO 6Ø
17Ø GOSUB 238Ø
18Ø GOSUB 241Ø: RUN
2Ø5 RESTORE 257Ø
21Ø FOR I=52Ø TO 567
215 READ B
22Ø VPOKE I,B
225 NEXT I
23Ø FOR I=576 TO 615
24Ø READ BA
25Ø VPOKE I,BA
26Ø VPOKE I+64,BA
29Ø NEXT I
8ØØ REM
81Ø R=FNR (5Ø)
82Ø IF R>5 THEN R=Ø
83Ø IF R>4 THEN R=3:RETURN
84Ø IF R>1 THEN R=2
85Ø RETURN
257Ø DATA 16,16,6Ø,126,255,189,
     231,231
259Ø DATA 16,56,84,16,56,84,146,
     16
261Ø DATA 8,2Ø,34,65,6,8,16,224
263Ø DATA Ø,48,72,132,2,Ø,Ø,Ø
2635 DATA 255,231,231,129,129,
     231,1Ø2,6Ø
2636 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
265Ø DATA 128,24Ø,255,252,143,
     128,128,128
267Ø DATA 64,24Ø,72,68,68,68,78,
     68
```

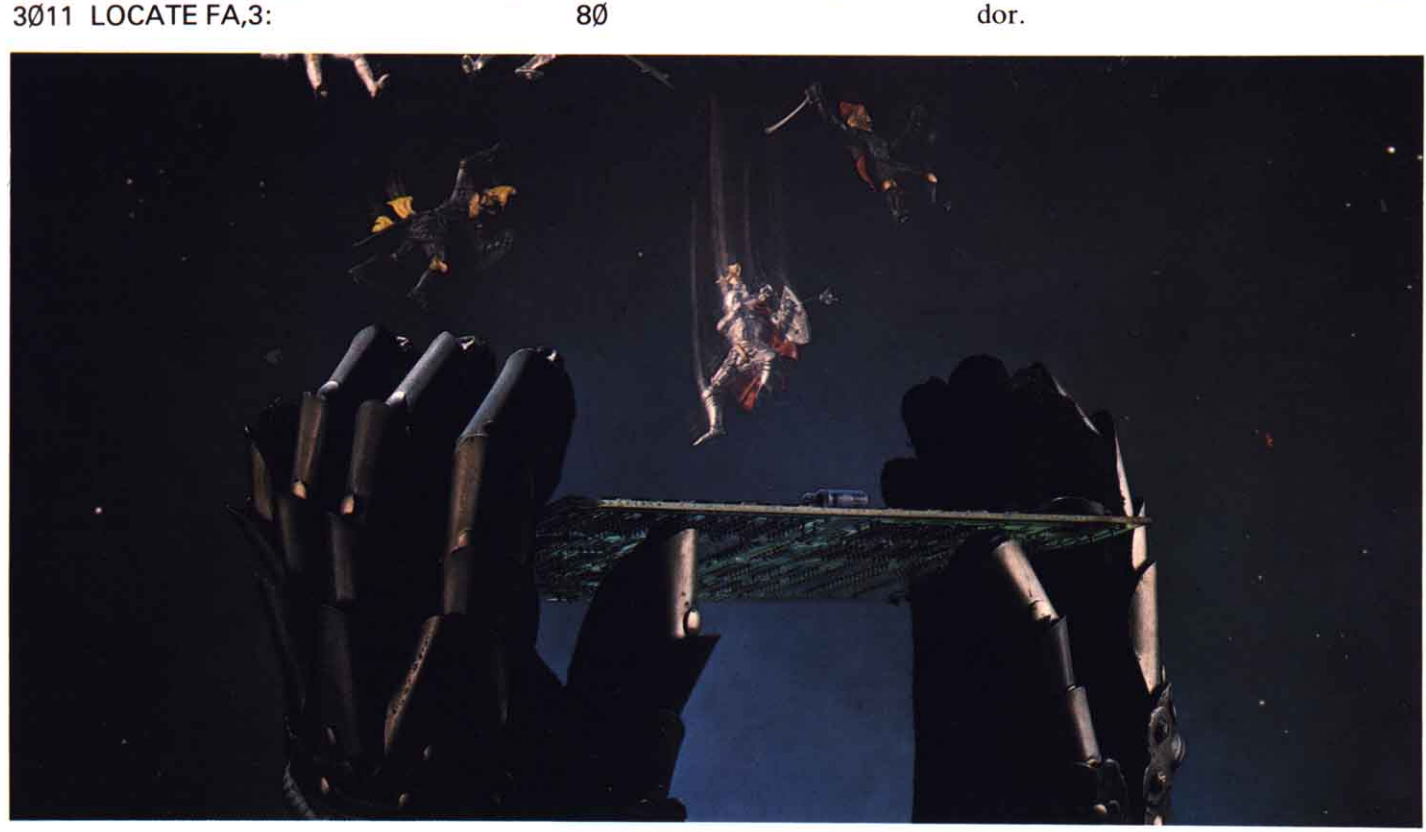
```
2690 DATA 255,231,231,129,129,
     231,102,60
2710 DATA 249,70,38,25,9,5,3,1
2730 DATA 1,2,4,8,16,160,64,160
2900 FOR FZ=1 TO 8
2905 T(FZ+8,7)=INT(T(FZ+8,7)
     *1.5)
2907 NEXT: RETURN
2910 FOR FZ=1 TO 8
2915 T(FZ+8,7)=T(FZ+8,7)*2
2917 NEXT: RETURN
2920 FOR FZ=1 TO 8
2925 T(FZ+8,7)=T(FZ+8,7)*3
2929 NEXT: RETURN
3000 GOSUB 3030
3001 LOCATE 5,1: PRINT"WAR
     GAMING"
3002 LOCATE 8,5: PRINT"1–
     FACIL"
3003 LOCATE 8,9: PRINT"2–
     NORMAL"
3004 LOCATE 8,13: PRINT"3–
     DIFICIL"
3005 LOCATE 8,17: PRINT"4–
     IMPOSIBLE"
3006 FOR FA=3 TO 19
3007 LOCATE 6,FA:
     PRINTCHR$(199)
3008 LOCATE 21,FA:
     PRINTCHR$(199)
3009 NEXT
3010 FOR FA=7 TO 20
3011 LOCATE FA,3:
     PRINTCHR$(199)
3012 LOCATE FA,19:
     PRINTCHR$(199)
3013 NEXT
3014 T2T=2
3015 ID T2T=1 THEN VPOKE
     8216,111: T2T=2: GOTO
     3020
3016 IF T2T=2 THEN VPOKE
     8216,1: T2T=1
3020 LET G2$=INKEY$: IF G2$="
     " THEN GOTO 3015
3021 IF ASC (G2$) < 49 OR ASC
     (G2$)>52 THEN GOTO 3020
3022 LET TT1=VAL(G2$)
3023 CLS: RETURN
3030 CLS: RESTORE 3604: FOR
     N=1 TO 29
3031 READ Q: Q$=Q$+CHR$(Q):
     NEXT N
3035 PRINT: FOR N+1 TO
     7:PRINTQ$: PRINT: NEXT N
3036 FOR N=1 TO 400: NEXT N
3040 CLS: RETURN
3200 FOR FD=1 TO 16
3210 LET T(FD,1)=2
3220 NEXT FD
3230 RETURN
3604 DATA 42,98,121,32,32,68
3605 DATA 65,78,73,69,76,32,67,
     65
3606 DATA 76,86,69,84,32,76,69,
     80
3607 DATA 69,75,72,73,78,69,42
     8192 ,192
```

INSTRUCCIONES DEL JUEGO

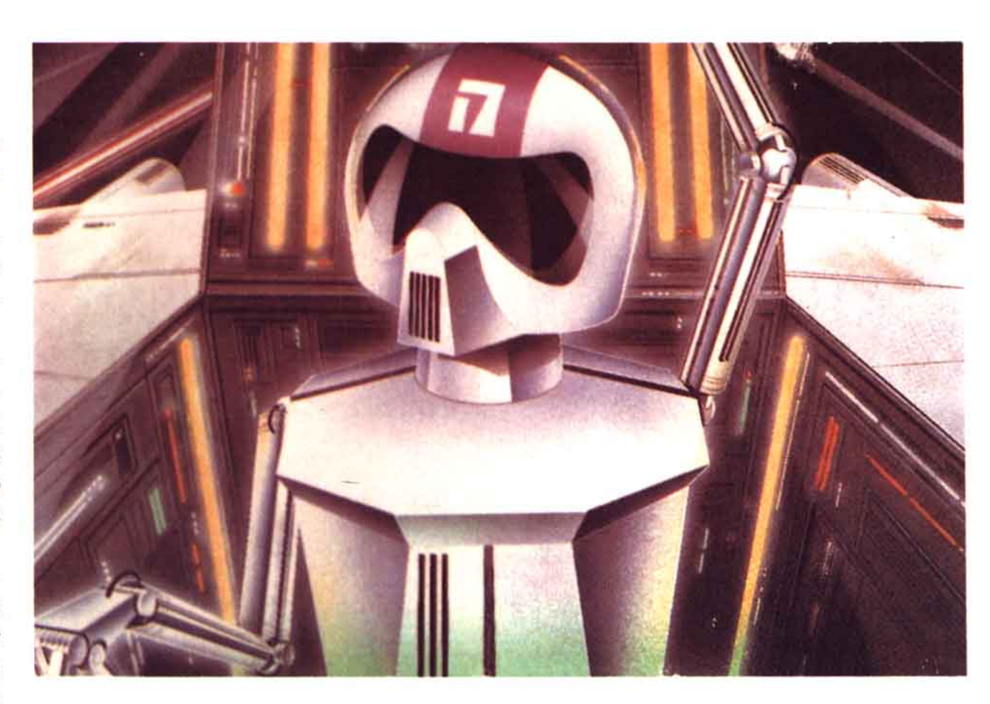
Inmediatamente que ejecutamos (RUN) el programa, el ordenador se pone a dibujar el mapa. Los símbolos del terreno aparecen en primer lugar, seguidos de las unidades oponentes y de la frontera.

Ahora es el momento de empezar a construir la estrategia. Una serie de símbolos aparecerán en la «ventana» del texto. Comenzando desde la unidad uno, el número de unidad y la descripción –por ejemplo, caballeros– junto con órdenes actuales, **Hacer alto,** tal vez. Al jugador se le pregunta **Cambio (S/N)?**.

Si la respuesta es S, se muestra un menú de órdenes opcionales: **Abrir fuego, Hacer alto** o **Moverse.** La opción de **Abrir fuego** sólo queda abierta para los arqueros; así pues, cualquier intento de hacer que otro tipo de unidad abra fuego, hará aparecer el mensaje «Ningún arco», y el ordenador esperará otra elección. Si se selecciona la opción de movimiento, el apunte **Qué camino?**, (N,S,E,W) hará su aparición, listo para la elección del jugador.



FIN DE UNA SERIE Y COMIENZO DE UNA NUEVA ETAPA



Con este coleccionable acabamos la serie que venimos publicando desde los inicios de la revista INPUT MSX.

Esperamos que haya sido de vuestro agrado y que hayáis profundizado en la sencillez del BASIC y en la programación estructurada.

Todos los programas que hemos venido presentando mes a mes han cubierto las expectativas más diversas: desde clásicos comecocos hasta juegos de aventura, y en este último número un cuadernillo especial dedicado a los *wargames* (o juegos de estrategia).

Acabamos un coleccionable suponiendo que el lector, un tanto hastiado del lento BASIC, gustará de probar nuevas delicias, como pueden ser los ensambladores y el lenguaje máquina que ello conlleva.

El próximo número de INPUT MSX abre una nueva etapa en la era de la programación: el código máquina, ese ser tan esperado y tan temido, que nos hará adentrarnos en un nuevo mundo más veloz y, a la vez, más absorbente. Su título será *RR contra el Doctor Rom*, y puntualizamos que no será ningún tipo de coleccionable sino un simple cuadernillo, independiente en su contenido de un mes a otro.

INPUT MSX

PRESENTA:

RR CONTRA EL DOCTOR ROM

INPUT

PRÓXIMAMENTE EN VUESTRAS PANTALLAS

BASADO EN UN GUION DE MICROSOFT S.A.
CON NEL MANEL
DAC DELA • TON LABIANDA • RAB MEGA en R.R. y
GERMAN LEPE QUINA en Doctor Rom
ESTRELLA INVITADA, MARGOT ERA
DIRIGIDO Y PRODUCIDO POR SAD SACK

CARTELERA

LOS MEJORES CUADERNILLOS DE CÓDIGO MÁQUINA JAMÁS VISTOS

«La trama es algo apasionante: secretos del BIOS descubiertos, agentes de ROM ejecutados por subrutinas fascinantes, artefactos del PPI activados por miembros del Área del Sistema... algo inimaginable hasta ahora en revistas del Standar...»

PEEK van POKE (Crítico de software)

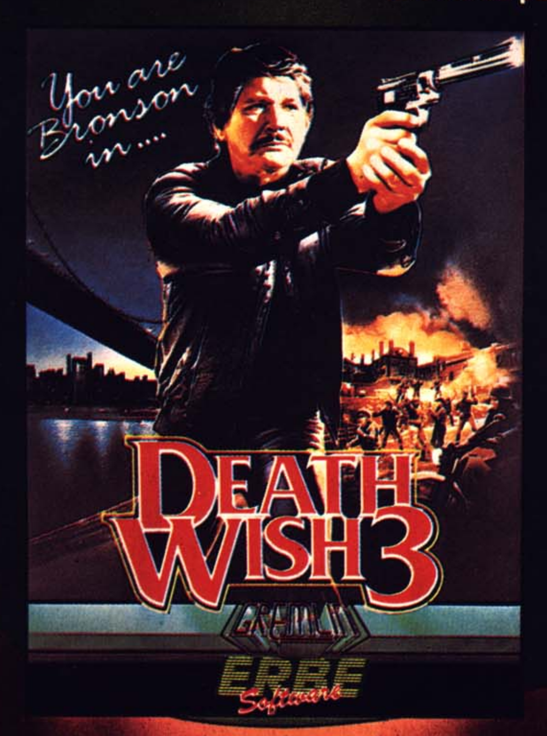
...EN SUMARIO A PARTIR DEL NÚMERO 17 DE INPUT MSX

«Espías occidentales han atravesado las barreras niponas en busca de confidencias sobre el ordenador definitivo: ¿Podrán conseguir sus objetivos y regresar sanos y salvos?... La VDP (servicios secretos visuales) no perdona errores».

X. FERR (Gran maestro del soft)

MANEJAR CON PRECAUCION

¡¡ATENCION!! Hay un explosivo par de novedades a punto de llegar a las tiendas. En buenas manos producirán horas de emoción, entretenimiento y acción. Mal manejadas pueden causar alteraciones nerviosas y taquicardias... Tú sabrás si tienes la confianza y habilidad necesarias para aceptar el desafío.





¡¡DINAMITA PURA!!





MAPA DE...

YIE AR KUNG-FU

Son muchos los meses que han pasado desde que este sensacional programa basado en las artes marciales apareciera en el mercado. Debido a vuestras insistentes peticiones de un mapa y consejos de cómo llegar hasta el final, este mes os ofrecemos ambas cosas.

Para empezar podemos dar un vistazo al mapa y ver cuáles son los enemigos a los que tendremos que enfrentarnos.



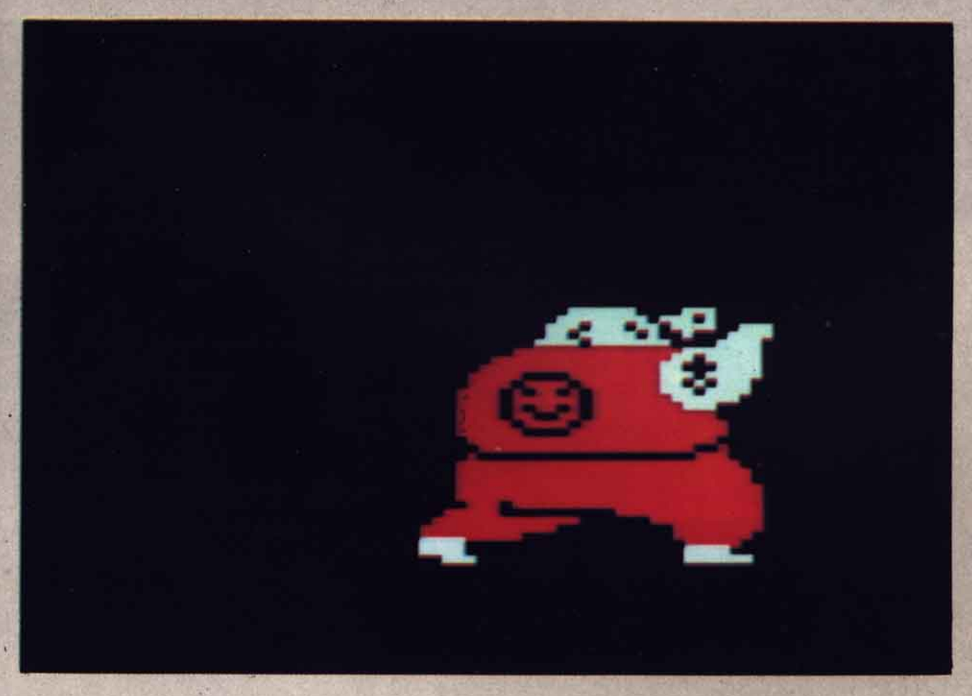
En primer lugar nos encontramos con POCHIN, voluminoso y fofo enemigo cuya arma principal es su mística coleta dorada. La forma ideal de reducir su poderío y vitalidad es alejarse lo

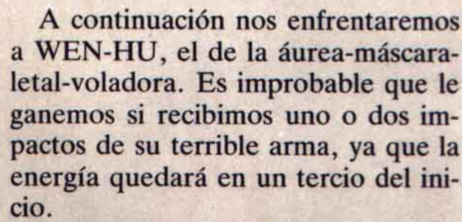


más posible de su persona, preparar un buen salto, golpear con precisión y retirarse antes de que la coleta entre en acción.

Después de vencer a este endemoniado enemigo tendremos un duelo con la enigmática vampiresa-diablesa oriental LANG-FANG, señora de los abanicos voladores. La mejor táctica para acabar con su poderío es acercarse lo más posible a ella, evitar los abanicos con saltos y golpes, y dar patadas (aunque ello no sea muy caballeresco) a diestro y siniestro.

YEN-PEI, el bolero mágico (bolero por las bolas paralizantes que tira) es quizá uno de los más difíciles contrincantes con los que nos tendremos que haber. No existe una táctica concreta para destruirlo; tal vez la más sencilla sea agacharse, esquivar las parálisis etéreas y atacar con los puños en su fuerte corpachón. El combate cuerpo a cuerpo es imprescindible para alcanzar el éxito.

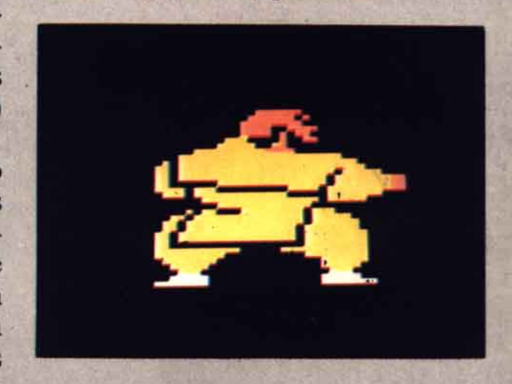




A continuación nos enfrentaremos a WEN-HU, el de la áurea-máscara-letal-voladora. Es improbable que le ganemos si recibimos uno o dos impactos de su terrible arma, ya que la energía quedará en un tercio del inicio.

La táctica primordial es destrozar ésta con una patada voladora o un buen golpe de mano. Atención al combate cuerpo a cuerpo y a sus tácticas de golpe y tentetieso.

MEI-CHIN, australiano por parte de tío, es el encargado de rebanarnos el pescuezo (digo la energía) con su afilado boomerang de acero inoxidable; mientras lo evitemos y, aproximándonos, le sacudamos como a una estera vieja, no hay problema.

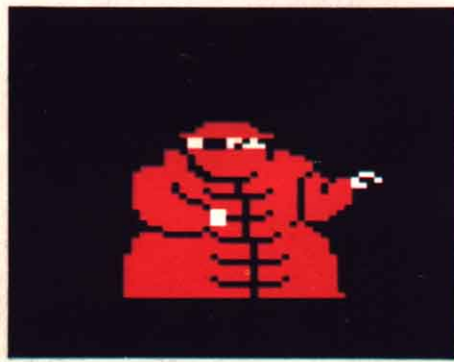




En la fase sexta del juego nos encontramos a otra mujer fatal. Fatal porque como te alcance con la navaja trapera te deja fino de energía. Para acabar con ella lo mejor es darle un impacto y largarse lo más rápido posible, así una y otra vez hasta vencerla.

En el penúltimo nivel, HAN-CHEN hará las delicias de todos los bomberos. Sus granadas explosivas, si nos tocan, descienden el nivel de vitalidad y conocimiento (pues nos dejan atontados) hasta un límite bajísimo. Huyamos de ellas como la peste y ata-



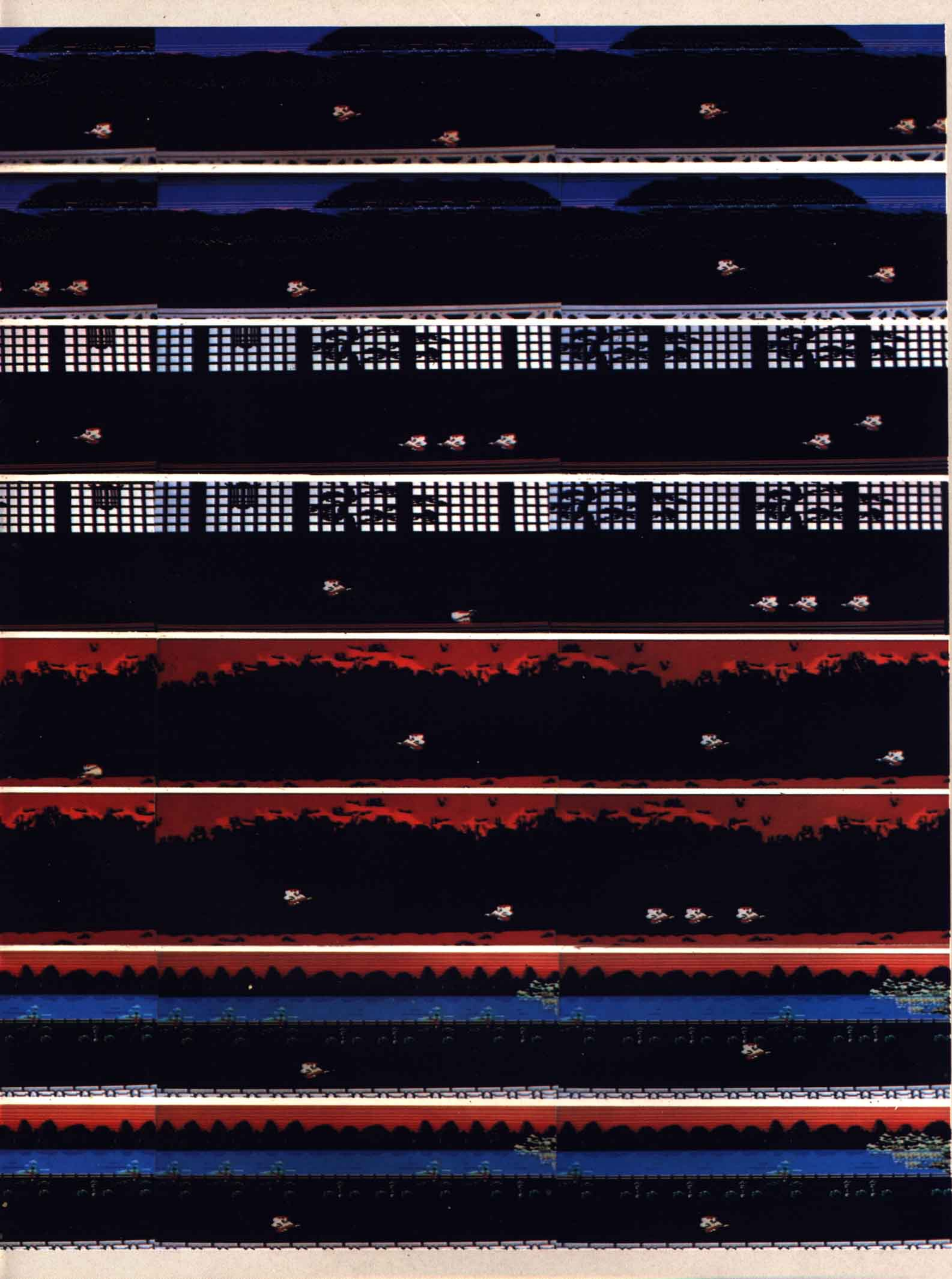


quemos fugazmente su parche ocular hasta que pase a mejor vida.

Y por fin llegamos al malvado señor y capo mafioso-oriental que nos ha mandado su escuela de sicarios por correspondencia: el señor de la muerte, LI-JEN. Sus rayos zeusinos son mortales, su celeridad inmensa, su maldad ominosa... Sólo existe un camino para alcanzar la vía del Zen: esquivar los haces de energía y golpear con rabia y cólera en los momentos de calma. Calma que antecede a la tragedia: uno de los dos será el ganador. No obstante, si seguís al pie de la letra nuestros consejos, el triunfo será vuestro.

Así que ya lo sabéis: ¡valor y al chino!

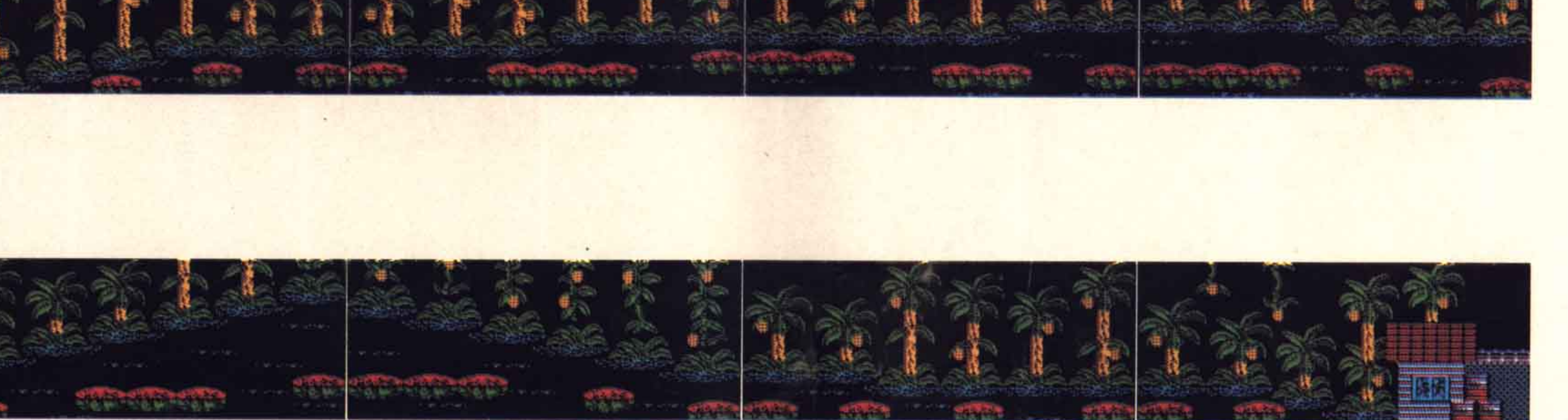


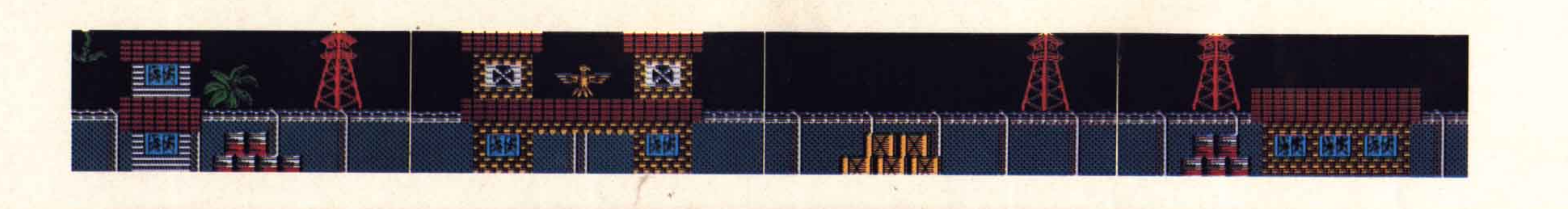


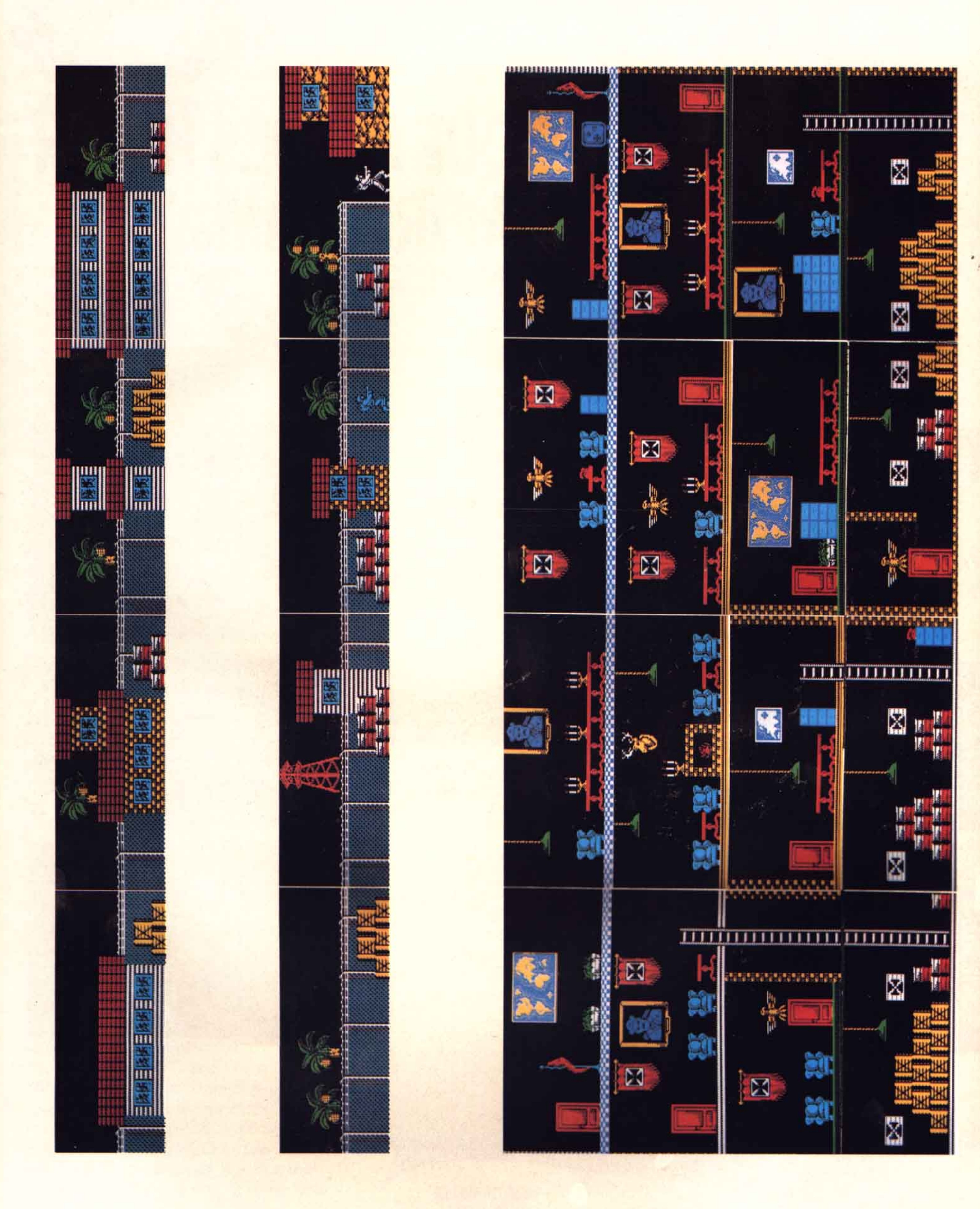


MAPA DE LA SEGUNDA PARTE DE









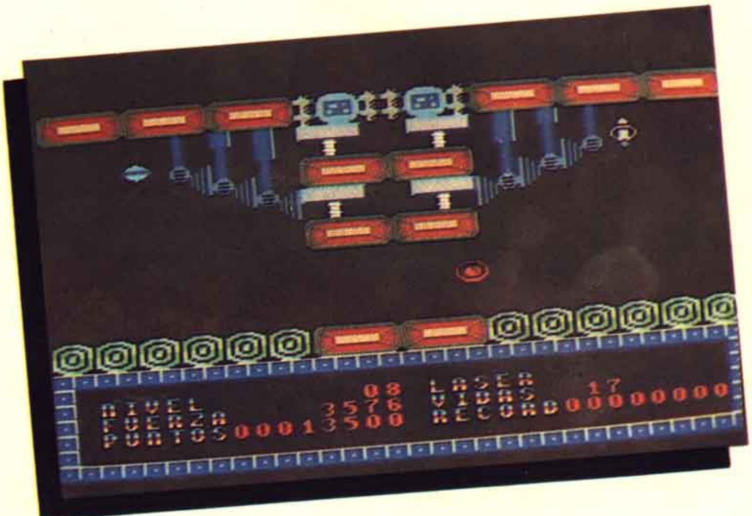
MAPA, CARGADOR Y POKES PARA... THE LAST MISSION

Para entender un juego y saber lo que se nos pide realizar en él, lo primero que debemos hacer siempre es leernos la trama o historia en la que se basa. Hay casas de software que no dan ninguna importancia a esta ambientación, en cambio otras la encuentran de vital importancia, y éste es el caso de los programadores de OPERA SOFT, quienes han inventado una bonita historia para su nuevo juego: THE LAST MISSION. Por esta razón, además de haceros entrega del mapa de este juego y de facilitaros algunas útiles pistas para completar con éxito vuestra misión, os vamos a hacer un pequeño resumen de la historia en la que se basa este arcade. Y esta historia es la siguiente:

Hace mucho tiempo que la Tierra cayó en su propia trampa, la trampa de la automatización. Los autómatas y máquinas inteligentes que en el pasado el hombre había fabricado para su propia satisfacción, ahora se volvían contra sus creadores. Era «La revolución de las máquinas». Una vez estos seres metálicos y sin corazón ganaron la batalla, decidieron acabar con todo vestigio de vida sobre la faz de la tierra y construyeron una gigantesca central, base energética de su poderío. Pero el hombre, que nunca se da por vencido, se refugió en el planeta Nova y envió al robot OR-CABE-3 a la tierra, donde consiguió robar los planos de defensa del enemigo. Ahora su misión es lograr escapar de la Gran Central, donde se encuentra, y volver a Nova para entregar los planos a los humanos.

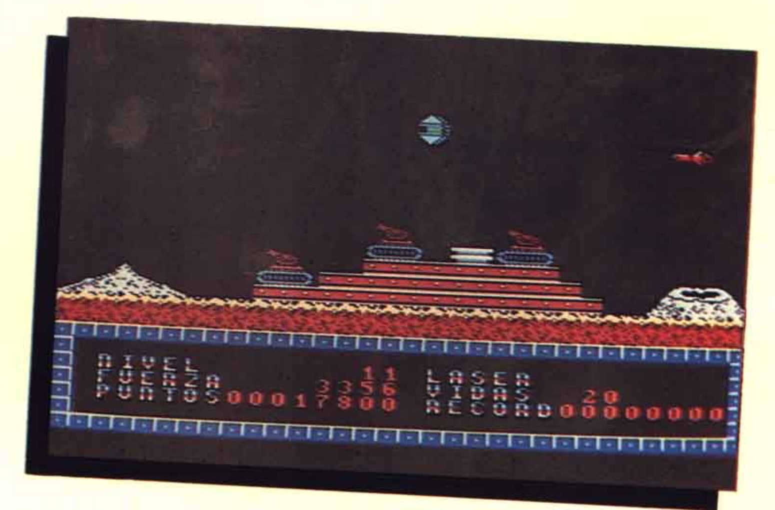
Bonita historia ¿no? Nos hemos saltado los detalles técnicos de cómo es nuestro robot OR-CABE-3 y de cómo salir de la nave para poder comentar a continuación, y de forma extensa, estos y otros importantes aspectos del programa. OR-CABE-3 es una original máquina compuesta por dos módulos bien diferenciados, la oruga locomotora y una cabeza propulsora. La oruga locomotora tan sólo avanza cuando encima suyo se encuentra acoplada la cabeza propulsora. En cambio, la cabeza propulsora puede desplazarse por todo el interior de la Gran Central, hasta que se le acabe la energía, que debe ir a reponer en la oruga (por ello que conviene que las dos partes que forman al robot OR-CABE-3 permanezcan lo más cerca posible). Además, la oruga locomotora no tiene ningún arma para defenderse de los ataques enemigos, y tan sólo la cabeza propulsora está armada con un potente láser. En un principio, OR-CABE-3 se encuentra en el nivel más bajo de la Gran Central, y para lograr salir de ella deberá subir hasta la superficie del planeta, habiendo superado todos los mecanismos de defensa que habitan en este magno edificio. Para poder ir subiendo nivel a nivel hasta la superficie es necesario juntar las fuerzas de las dos partes de nuestro robot. La cabeza propulsora deberá llevar a la oruga locomotora hasta el ascensor que se encuentra en un extremo del nivel por donde circulan, y de este modo todo el robot OR-CABE-3 podrá subir de nivel. Si pasamos de nivel tan sólo con la cabeza propulsora, cuando la destruyan o se le acabe la energía de vuelo, aparecerá de nuevo donde se encontrara la oruga locomotora. Por ejemplo, si logramos llevar a la oruga locomotora hasta el cuarto nivel y allí nos destruyen, reanudaremos nuestra misión en el cuarto nivel, mientras que si hubiéramos dejado la oruga en el nivel anterior, la reanudaríamos en el 3.º. Además de esta ventaja de hacer avanzar conjuntamente todo el robot OR-CABE-3 y no hacerlo por partes hay otras, entre las que destaca la siguiente: dentro de la Gran Central también actúa la fuerza de gravedad, de modo que la cabeza propulsora debe estar en suspensión o arrastrándose por el suelo, y aquí está el problema, ya que arrastrarse gasta menos energía y es mucho más cómodo, pero el sistema defensivo de la Gran Central ya ha previsto este detalle y ha colocado minas en el suelo, y a veces hasta en el techo, de modo que si la cabeza propulsora choca con ellas se destruirá. En cambio la oruga locomotora es inmune a estas minas, y por esto es aconsejable, siempre que se

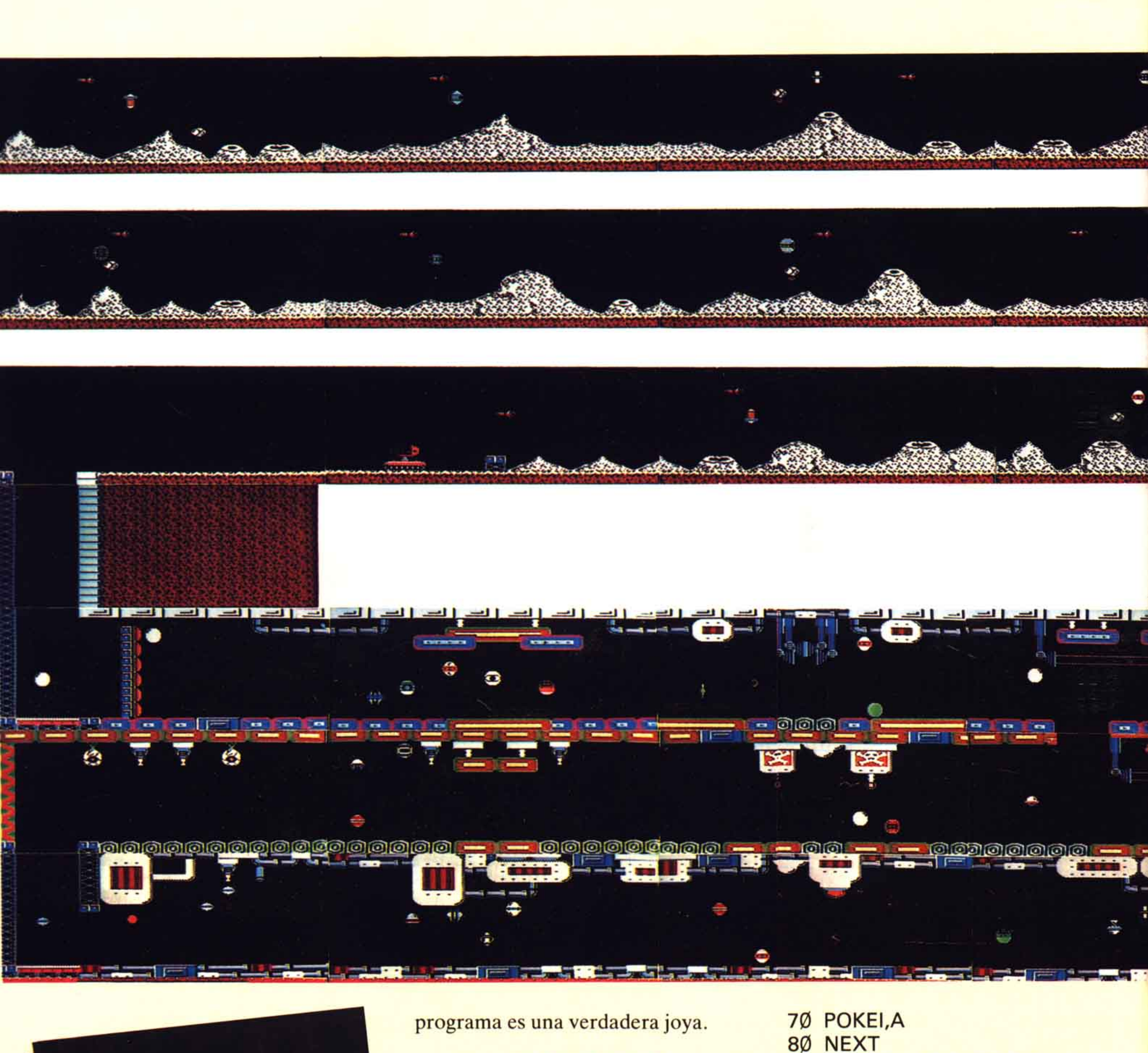




pueda, usar conjuntamente nuestro robot para avanzar por estas zonas tan peligrosas. No obstante, a veces será inevitable tener que separar las dos partes de OR-CABE-3, por ejemplo en el caso de encontrarnos con los muros bicolores. Estos muros que se interponen en el camino de nuestro robot hacia el ascensor que le hará subir de nivel, tan sólo pueden ser destruidos por uno de sus dos lados, el de color rojo, y éste siempre se encuentra en la cara opuesta a la que está OR-CABE-3. La única solución consiste

en separar los dos componentes del robot y hacer que la cabeza propulsora viaje por las interioridades de la Gran Central hasta llegar al otro lado del muro. Así podremos destruirlo y reacoplar las dos partes del robot para que éste continúe su camino hacia la superficie del planeta. En los niveles inferiores esto nos va a ocurrir una vez por nivel, pero a medida que avancemos este fenómeno se va a convertir en un verdadero tormento. Por suerte, OR-CABE-3 cuenta con vuestra inestimable ayuda. En resumen, el

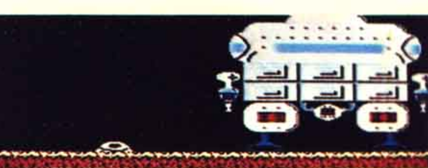


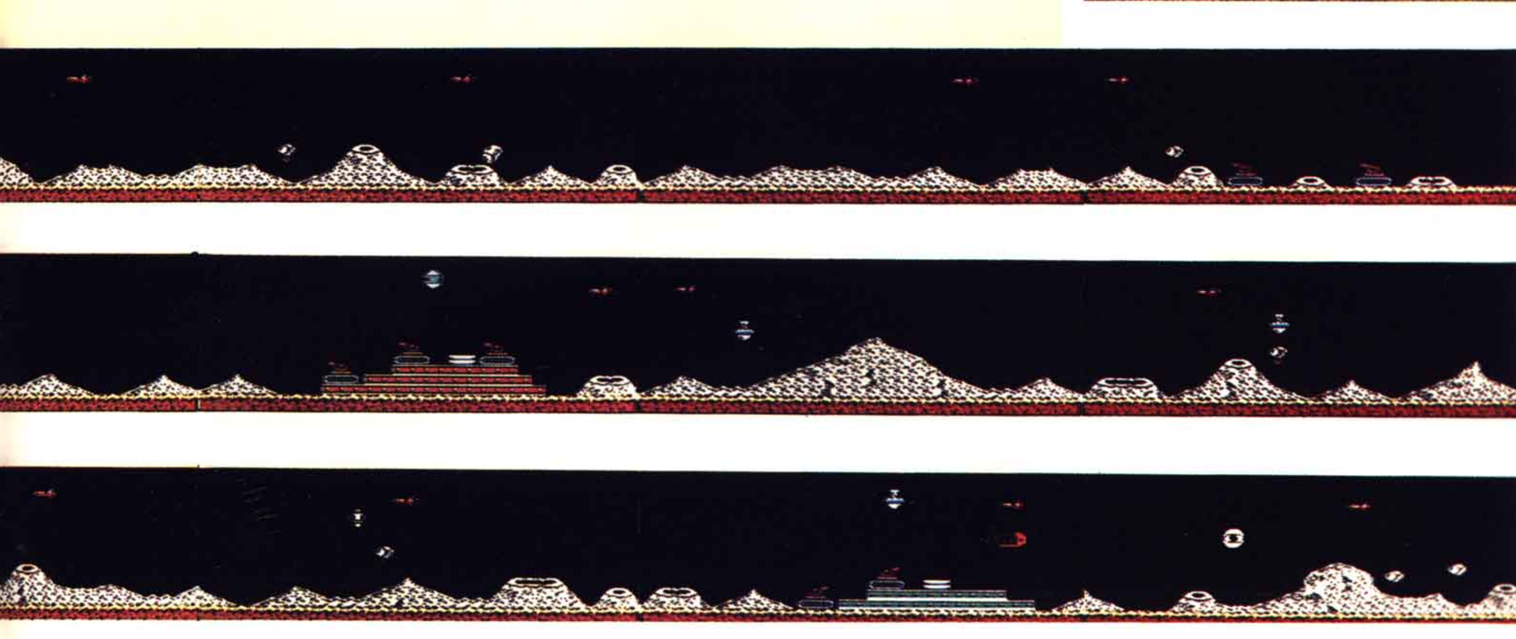


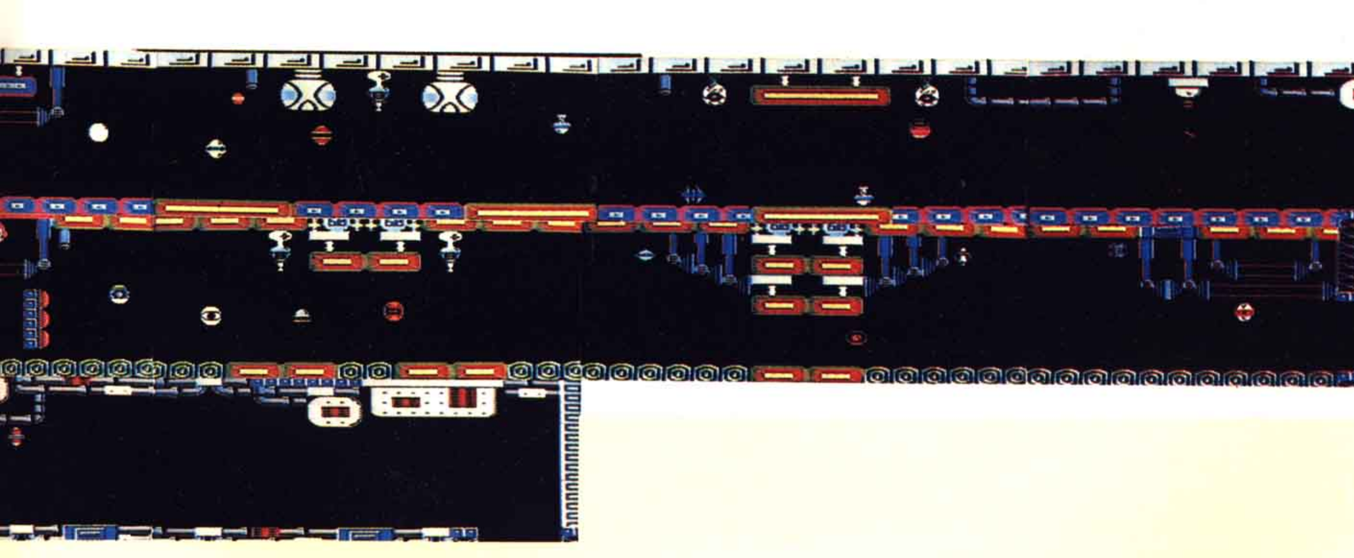
programa es una verdadera joya.

EL CARGADOR

```
1Ø  REM----INPUT MSX----
2Ø  REM----POR------
3Ø  REM----JUANMA-----
4Ø  FORI=&HEØØØTO&HEØ11
5Ø  CLS:KEYOFF
6Ø  READA$:A=VAL("&H"+A$)
7Ø  POKEI,A
8Ø  NEXT
9Ø  POKE&HDD6C,Ø:POKE
    &HDD6D,&HEØ
1ØØ LOCATE2,6:PRINT"VIDAS  IN-
    FINITAS--------1"
11Ø LOCATE2,8:PRINT"ENERGIA
    DE VUELO INFINITA---2"
12Ø LOCATE2,1Ø:PRINT"ENERGIA
    DE LASER INFINITA---3"
```







```
130 LOCATE2,13:PRINT"EMPE-
    ZAR LA CARGA-------Ø"
140 LOCATE4,21:PRINT"SELEC-
    CIONA LAS OPCIONES"
150 A$=INPUT$(1)
160 A=VAL(A$):IFA$="Ø"THEN 190
170 IFA<Ø OR A>3THEN15Ø
180 B(A)=1:GOTO15Ø
190 BLOAD"CAS:"
200 FORI=1 TO3
210 READA$(I):A=VAL("&H"+A$(I))
220 IF B(I)=1THENPOKEA,Ø
230 NEXT
240 POKE&HDD6C,Ø:
    POKE&HDD6D,&HEØ
250 CLS:DEFUSR=&HDDØØ:
    A=USR(Ø)
260 DATA 3E,3D,32,4C,47,3E,3C,32,
    EC,46,3E,34,32,BE,42,C3,Ø3,4Ø
270 DATA EØØ1,EØØ6,EØØB
```

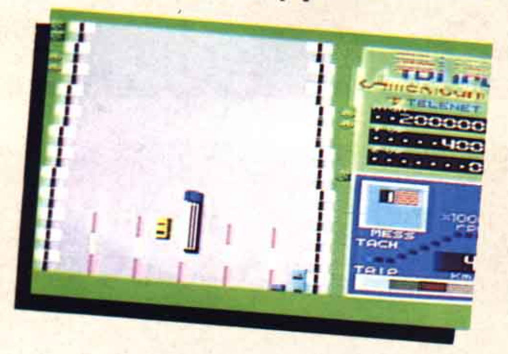
• OPERA SOFT ▲ CASSETTE/DISQUETTE ■ ARCADE

ANIMACION	8
INTERES	8
GRAFICOS	8
COLOR	8
SONIDO	8
TOTAL	40



AMERICAN TRUCK

Las carreteras están llenas de peligros y conductores temerarios que harán nuestra travesía por tierras americanas más amena y entretenida. Y es que en AMERICAN TRUCK nosotros actuamos bajo el papel de conductores de camión. Con nuestro vehículo nos deberemos abrir paso por las infernales carreteras, y por cada coche



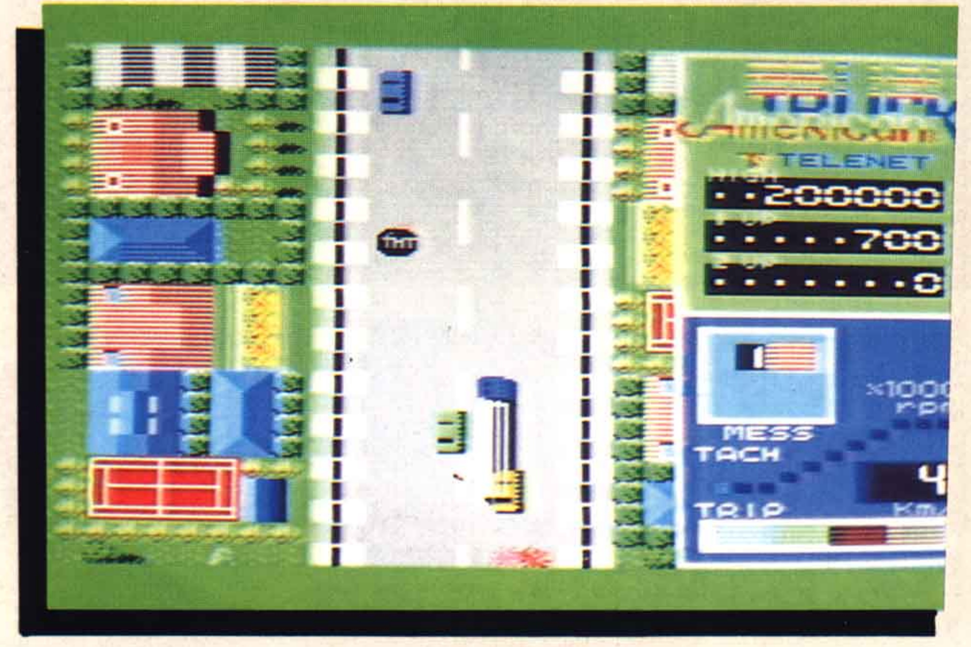
que hagamos que se estrelle contra la calzada se nos recompensará con un número variable de puntos. Pero nosotros también deberemos estar atentos a los peligros de la carretera, ya que de lo contrario también vamos a tener que abandonar nuestro camino, y es que ésta es una carretera a vida o muerte donde todo está permitido. Así que si queremos ganar deberemos saber aprovechar la mayor fuerza y envergadura de nuestro camión, en la temeraria lucha contra los otros conductores, que por suerte llevan pequeños y débiles coches que fácilmente podemos desestabilizar.

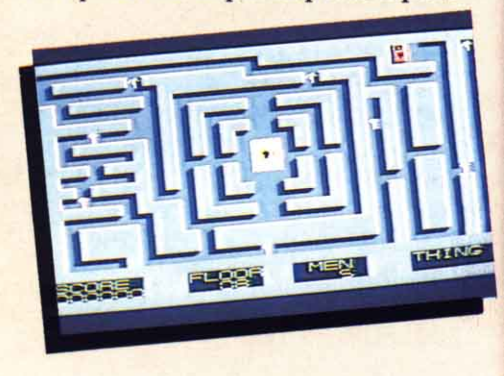
Éste es un juego en que no hacen falta pistolas, ni granadas, ni metralletas para acabar con el enemigo: tan sólo nos son necesarios unos buenos reflejos y la suficiente habilidad como para esquivar los peligros de la carretera. Todos luchamos contra todos y nadie pisa el freno para dar paso a otro vehículo; es más, si acelerando lo destruimos más vale que no nos lo pensemos dos veces y pisemos el acelerador a fondo. Ganará quien conduzca mejor, pero también quien sea más temerario y tenga menos escrúpulos.

MIDNIGHT BUILDING

Nos encontramos en la octava planta de un edificio, pero algo malo debemos haber hecho porque tras nosotros van los tres vigilantes del inmueble. Para lograr escapar de esta encerrona tenemos que lograr acceder a la planta baja y salir del edificio. Desgraciadamente, las puertas de todas las plantas se encuentran cerradas, de modo que antes de salir de una planta debemos recoger todas las llaves que por allí se encuentren y seguidamente, sin tropezar con ninguno de los vigilantes, abrir una puerta y bajar al siguiente nivel.

En MIDNIGHT BUILDING cada planta o nivel viene representado por una única pantalla gráfica hecha en forma de laberinto. En este laberinto nos encontramos nosotros, los vigilantes del edificio que intentarán cazarnos y las llaves que nos permitirán escapar de cada nivel. Nuestro objetivo es recolectar todas y cada una de estas llaves, abrir la puerta y escapar. Pero como los vigilantes no nos van a dejar en paz, estamos armados con una pistola que nos permitirá realizar cuatro disparos en cada planta, los cuales tan sólo dejarán momentáneamente paralizados a los vigilantes. Otra dificultad adicional es que nos encontramos en la oscuridad de la noche y por tanto nuestra visibilidad no es completa, es decir, no vemos toda la planta, sino tan sólo un rectángulo, en el centro del cual estamos nosotros, y que nos sigue en nuestros desplazamientos. Este hecho es un verdadero handicap, ya que al no poder ver la planta por completo





no sabemos dónde se encuentran las llaves, no sabemos por dónde se mueven los vigilantes, y no sabemos si vamos por buen o mal camino, de manera que más de una vez vamos a acabar atrapados en un camino sin salida, por lo cual es conveniente reservar las cuatro balas de nuestra arma.

Éste es un programa basado en una sencilla idea, pero gracias a que ha sido llevado a cabo con elegancia y buen gusto, sin intentar dar más de lo que se podía, debemos decir que su calidad es más que aceptable, y que tiene como mejor arma su alto poder adictivo. Que no os coja desprevenidos.

INCA

INCA es uno de tantos juegos del estilo de JET SET WILLY, y es que este programa descubrió un filón de oro para los programadores de juegos para microordenadores. Por tanto ya sabéis todos de qué tipo de juego se trata.

En un laberinto de pantallas llenas de peligrosos enemigos y abismos mortales e insondables se encuentran colocadas diferentes piezas que debemos ir recolectando, en concreto 150 lingotes de oro. Pero para alcanzarlos deberemos ir con mucho cuidado y lograr superar todas las trampas que encontraremos hasta llegar al dorado botín.

Los enemigos se desplazan según repetitivos movimientos, los cuales debemos estudiar a conciencia para



superarlos en el momento más oportuno y de menor peligro, aunque siempre acabaremos por cometer algún error, terminando de este modo con alguno de los cinco exploradores con cuyo concurso podremos completar felizmente la misión.

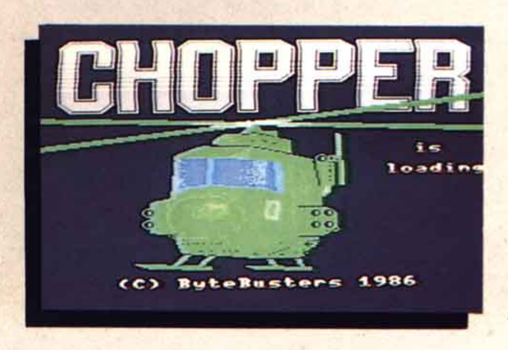
Si acabáis por adquirir este programa, si es que algún día alguien se digna a importarlo, no os podréis ver decepcionados por vuestra compra, ya que está bien claro lo que vais a buscar: un juego en que la concentración y la rapidez con los mandos de control sean las claves del éxito o el fracaso en la misión que se os pide.

Y por lo que respecta a la calidad del programa, os podemos informar que es más que aceptable, ya que al tratarse de un tipo de juego del que hay muchas versiones en el mercado, tan sólo la calidad puede marcar diferencias perceptibles entre unos y otros.



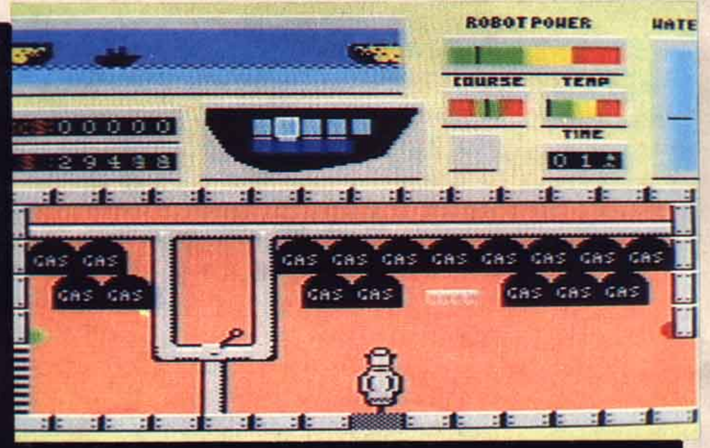
CHOPPER 2

Podemos decir que este juego de BYTEBUSTERS es algo más que un matamarcianos, ya que además de intentar acabar con unos terroríficos enemigos también tenemos una misión a realizar. El programa se apoya en una representación gráfica en tres dimensiones, de manera que nuestra nave puede desplazarse libremente en seis direcciones. Esta representación tridimensional nos recuerda mucho a la efectuada por SEGA en su programa ZAXXON, y es que los dos programas son similares. En CHOPPER, comandamos un helicóptero con el que debemos rescatar a supervivientes que hay en una base militar. Como por ella también sobrevuelan las naves enemigas, nos veremos involucrados en más de un combate a vida o muerte, pero por suerte casi siempre saldremos victoriosos. Una dificultad adi-



cional es la necesidad de reponer combustible antes de agotar completamente las reservas, ya que de lo contrario perderemos el control del aparato y nos estrellaremos contra el suelo. Y hablando del suelo, en él se encuentran instalados algunos misiles antiaéreos que se dispararán cuando pasemos por encima de ellos, de modo que no nos va a quedar otra alternativa que esquivarlos antes de que nos derriben. CHOPPER es un juego de alta dificultad en el que uno debe invertir muchas horas antes de lograr los primeros éxitos. Y es por esto que nunca nos va a sobrar tiempo para fijarnos en la calidad gráfica, que es muy buena si pensamos en la dificultad de hacer un programa de estas características.





SAILOR

A veces llegan hasta nosotros programas llenos de originalidad e inventiva, y SAILOR es un claro exponente de este tipo de juegos. La trama es la siguiente: nosotros estamos en un viejo barco de carga que poco a poco se va inundando, y para solucionar este problema, y otros que ya contaremos más adelante, disponemos de tres robots y un gráfico que representa al barco con sus siete compartimentos, tres de bodega y otros cuatro de cubierta. Una vez localizado el compartimento inundado nos dirigimos hasta él con unas placas que se encuentran en diferentes partes del barco, tapamos la grieta que hay en el suelo, y luego bombeamos el agua que había entrado. Pero seguro que otra sala ya está medio inundada y debemos correr hasta ella y repetir de nuevo toda la operación. Los otros problemas con que nos vamos a encontrar es que el barco cambie de rumbo y debamos volver a la sala de mandos para rectificarlo, o que la temperatura de los motores aumente considerablemente y debamos rebajarla. Y todo esto lo tenemos que hacer vigilando la energía del robot que estemos utilizando, ya que si abusamos de uno de ellos explotará irremisiblemente, y de seguir así nos podemos quedar sin quién nos haga el trabajo.

SAILOR además de ser un juego original es divertido y nos llega a poner los pelos de punta, y los nervios se disparan como en ningún otro juego. Y si a todo esto le unimos una presentación gráfica de calidad más que aceptable nos daremos cuenta de que no siempre es necesario pensar en matar marcianos para pasar momentos agradables jugando con nuestro ordenador MSX.

GALAXIA

GALAXIA es otro de tantos y tantos juegos que basan su atractivo en una nave intergaláctica que debe matar a tantos enemigos como pueda, de manera que la puntuación alcanzada por el jugador que comanda dicha nave sea la más alta posible, y de este modo pueda escribir su nombre en la hoja de récords. ¿Pero qué diferencia a este juego de otros de sus mismas características? Pues en primer lugar que la batalla se desarrolla en un plano horizontal, en contra de lo que nos tienen acostumbrados la mayoría de programadores. En segundo lugar que nuestra capacidad de ataque es poca, ya que mientras haya un misil en pantalla no podemos lanzar otro, y esto nos pone las cosas muy difíciles. El tercer aspecto a destacar de este juego es que existe una peculiar fuerza gravitatoria que nos hace caer no hacia la parte inferior de la pantalla, sino que





esta vez la nave se dirige inexorablemente hacia el lado izquierdo, extremo opuesto al margen por donde aparecen nuestros enemigos. También destacan en GALAXIA la posibilidad de escoger entre tres diferentes velocidades para el desarrollo de la partida, y la gran variedad de enemigos a los que nos tenemos que enfrentar. Estos enemigos se agrupan por niveles, de manera que cada nivel tiene su propio ejército de enemigos. Cuando acabamos con ellos o llegamos a esquivarlos a todos pasamos de nivel, y cambian las caras quienes se enfrentan a nosotros. Esta variedad hace que el programa se haga más entretenido y ameno. Ahora ya podéis apuntar un nuevo aspirante a fichar para vuestra colección de este tipo de bélicos arcades.



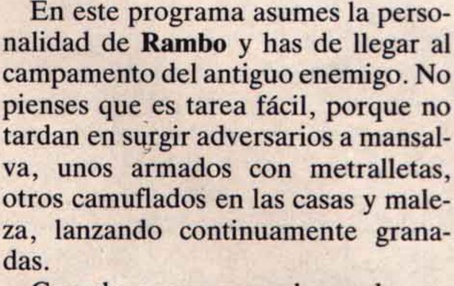
RAMBO

El título de la película **Acorralado** es probable que os suene a muchos.

Pero su segunda parte, **Rambo**, está a la orden del día.

Una comisión especial del Congreso de los Estados Unidos quiere demostrar que ya no quedan prisioneros de guerra norteamericanos en Vietnam y envían a **Rambo** en la esperanza de que no encuentra evidencias que demuestren lo contrario.

Rambo surge en la selva vietnamita portando únicamente tres armas: puñales, pistola y bomba, además de una cámara fotográfica que le servirá para aportar el documento gráfico si encuentra pruebas.



En este programa asumes la personalidad de **Rambo** y has de llegar al campamento del antiguo enemigo. No pienses que es tarea fácil, porque no tardan en surgir adversarios a mansalva, unos armados con metralletas, otros camuflados en las casas y maleza, lanzando continuamente granadas.

Cuando avanzas por la senda que conduce a un campamento hallarás un templo. En él localizarás una metralleta abandonada, que podrás recoger, pasando a formar parte de tu equipo. El camino es hostil y cada bala que te alcance disminuye tu fuerza. Cuanto mayor sea tu progreso, más difícil será mantenerte con vida.

Sigue tu rumbo hacia el norte y no tardarás en ubicar las vallas del campo de concentración, con sus torretas y guardianes armados. Sólo hay un acceso de entrada y verás que en la parte derecha del campamento existe un punto por el que puedes aproximarte a la valla, para situar una bomba que abrirá el boquete de acceso a la inexpugnable valla.

Una vez en el interior, mantener tu integridad se complica más cada vez. Ten por seguro que toparás con un sufrido prisionero atado. Corta las cuerdas que le aprisionan y llévalo contigo hasta el helicóptero que te aguarda para emprender la retirada. La presentación inicial de la versión en cinta sorprende agradablemente cuando la pantalla aparece inundada por una imagen digitalizada del cartel utilizado para la promoción de la película, mientras suena una sensacional melodía basada en el tema de la película. ¿No habrá un organista en el interior del televisor?

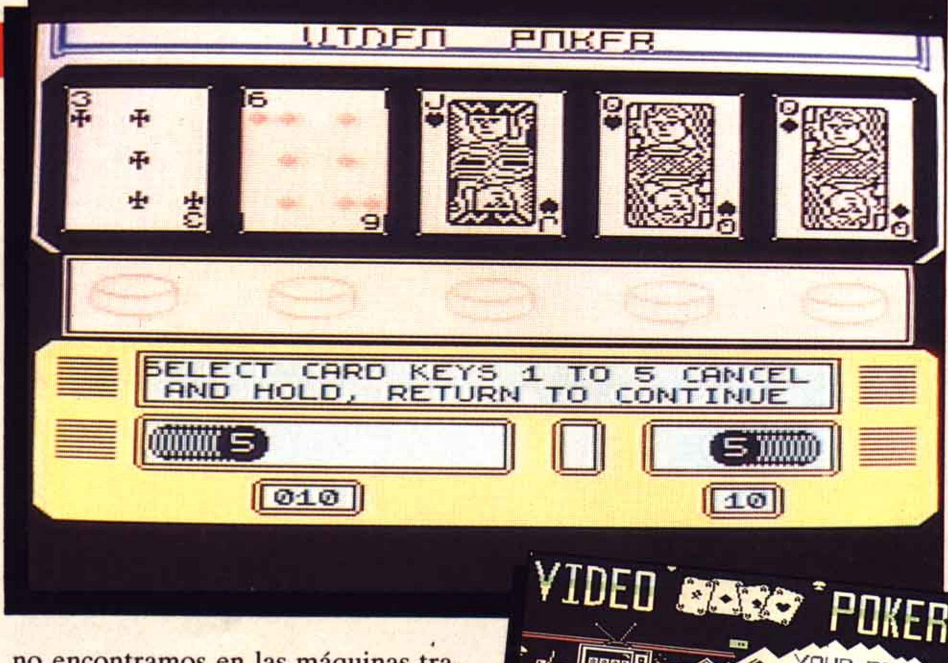
Está por demás aclarar que los gráficos son de una calidad inmejorable, sumándose a ello el hecho de disponer de toda una discoteca con la música apropiada para las distintas fases del juego.

El entretenimiento está asegurado con esta versión informatizada de **Rambo**.

VIDEO POKER

• MASTERTRONIC ▲ CASSETTE ■ POKER

Como su nombre indica VIDEO POKER es un programa basado en el famoso juego de cartas, pero esta vez no vamos a tener por contrincante a una bella dama, como sucedía en los strip-poker de SAMANTHA FOX STRIP y LAS LUCES ROJAS DE AMSTERDAM, sino que ahora el juego se ha planteado como si se tratara de una máquina tragaperras. En primer lugar se nos darán cinco cartas, pero boca abajo sin que podamos verlas realizando una apuesta, a continuación podremos girar las cartas y ver cuál es nuestro juego. Antes de acabar la partida podemos cambiar las cartas que deseemos, y una vez realizado dicho envite habrá terminado la jugada. Según cuales sean nuestras cartas finales se nos recompensará de un modo u otro. De esta manera si hemos conseguido una pareja recuperaremos el dinero apostado o si hay una doble pareja se nos premiará con una cantidad de dinero doble a la apostada inicialmente. Ahora si aún nos queda dinero y así lo deseamos podemos jugar de nuevo. Se barajarán nuevas cartas y podremos realizar nuevas apuestas. Un atractivo, que por desgracia no encontramos en las máquinas tragaperras de la calle, es la posibilidad de escoger entre cinco niveles diferentes, de manera que cuanto más elevado es menos posibilidades existen de realizar buenas jugadas. Otro acierto del programa es su sencillez de uso y que, aunque en inglés, siempre hay un mensaje en pantalla donde se nos dice cuáles son los movimientos que podemos realizar. También hay, representada en forma de gráfico, la cantidad de dinero que poseemos y la cantidad de dinero apostado. Y si interesa podemos acceder a una pantalla complementaria donde se indican todos los premios y cómo son pagados.

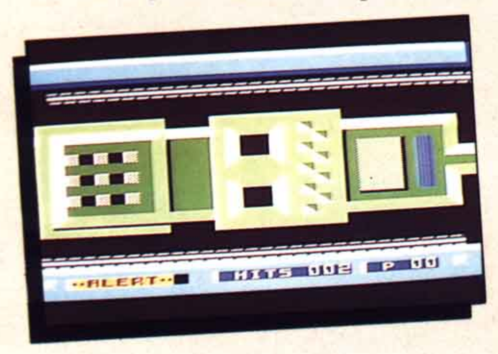


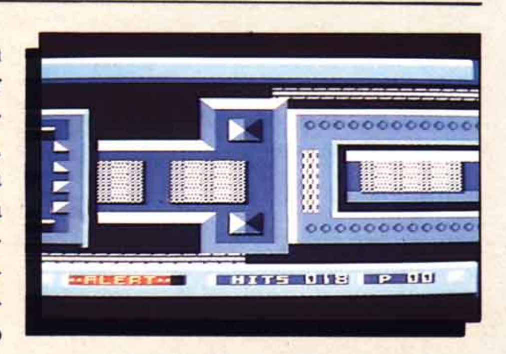
ANIMACION	5
INTERES	5
GRAFICOS	5
COLOR	4
SONIDO	3
TOTAL	22

FIRE HAWK

• MASTERTRONIC ▲ CARTUCHO ROM ■ JUEGO

¿Qué clase de juego o programa puede esconderse bajo el nombre del halcón de fuego, sino un matamarcianos o masacra-venusianos, como gusta decir nuestro redactor jefe? Pues bien, este arcade se caracteriza sobre todo por el movimiento de la nave que nosotros gobernamos. En primer lugar, nuestra nave avanza gracias a un *scroll* horizontal que la hace desplazar de izquierda a derecha, y que podemos acelerar o retardar, pero además, hay un movimiento vertical que lleva a nuestra nave de la parte superior a la inferior de la pantalla como si estuviera bajo los efectos de un campo gravitatorio. Y cómo no, nuestra nave también puede disparar proyectiles, pero no tantos como sería de desear, ya que mientras haya uno en pantalla no podemos lanzar el siguiente, y esto reduce considerablemente nuestra capacidad de ataque. Por ello, debemos afinar mucho el tiro e intentar que cada disparo se convierta en diana para destruir el mayor número de enemigos posibles. Éstos siempre aparecen por el lado derecho de la pantalla y seguidamente se desplazan hacia nosotros, dejándonos tan sólo dos caminos, destruirlos o esquivarlos. Mientras sea posible, os aconsejamos la primera solución. Un detalle importante: si nuestros enemigos quieren destruir nuestra nave, lo deberán hacer como kamikazes, destruyéndose a sí mismos.



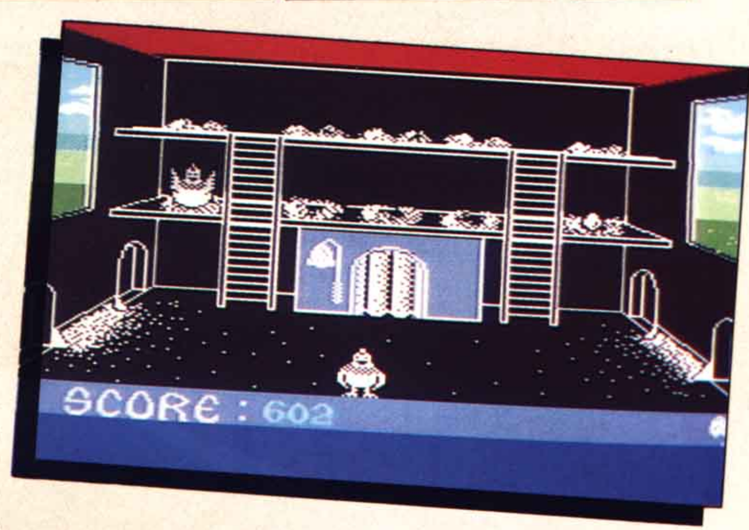


ANIMACION	3
INTERES	5
GRAFICOS	5
COLOR	5
SONIDO	4
TOTAL	22

CHICKEN CHASE

• BUG-BYTE ▲ CARTUCHO ROM ■ JUEGO

¿Nunca habéis pensado en convertiros en gallo de corral y tener todas las gallinas a vuestro alcance? Pues ésta es vuestra oportunidad. En CHICKEN CHASE seréis un apuesto pollo tomatero que tiene como misión salvaguardar a las gallinas y polluelos del gallinero. Pero ésta no va a ser tarea fácil. Las gallinas irán poniendo huevos sobre la suave paja del corral y luego se irán, de manera que seréis vosotros los responsables de su custodia. La espera, hasta que se rompa el huevo y salga de él un lindo polluelo, se hará larga y tensa. Existe el peligro de que las alimañas que viven en el bosque de alrededor vengan a robar estos huevos, y si no sois capaces de echarlos fuera se los comerán, y la gallina, al darse cuenta de lo ocurrido, os dará un buen escarmiento. Los peores enemigos son las comadrejas y las ratas, que entrarán sigilosamente para saciar su voraz apetito aprovechando cualquier despiste. Por ello, siempre deberéis estar atentos a sus inesperados ataques. Pero no todo iban a ser inconvenientes sino que también hay aspectos positivos, como son los gusanos que de tanto en tanto asoman la cabeza por el suelo del corral. Si logramos comérnoslos, además de alimentarnos un poco para estar fuertes para la lucha, se nos recompensará con una substanciosa cantidad de puntos que bien seguro nos ayudarán a alcanzar una buena posición en el ranking de puntuaciones.



Éste es un entretenido juego que se basa en una original idea, que además ha sido llevada a cabo con gracia y buen gusto. Y además con divertidos detalles, como la soberana paliza que nos pega la gallina que ha puesto el huevo cada vez que éste se extravía de un modo u otro.

ANIMACION	6
INTERES	5
GRAFICOS	5
COLOR	5
SONIDO	4
TOTAL	25

ZOOT

• BUG-BYTE ▲ CARTUCHO ROM ■ LABERINTO

Éste es un original juego de BUG-BYTE en el que debemos ayudar a Zoot, protagonista de la historia, a recuperar sus canicas. Éstas se le cayeron por una gruta y ahora se encuentran en las profundidades de un misterioso planeta, dentro del cual nos vamos a encontrar con los Goppa, los Bodkin, los Cheeky y los Grimbo. Todos ellos van a intentar que fracasemos en nuestro intento de recuperar las canicas de Zoot, y por eso él está muy agradecido de que le prestemos nuestra ayuda. Para defendernos de estos horribles monstruos de las entrañas del planeta tan sólo contamos con la propia fuerza de Zoot, en concreto con su gancho de derecha. Zoot es un tipo fuerte y para eliminar a los Goppa, los Bodkin, etc., se basta con sus puños. Si les llega a conectar un buen directo, éstos se mueren, ¡Pero cuidado! Ya que si los toca con cualquier otra parte del cuerpo que no sea su letal puño será él quien sucumba.

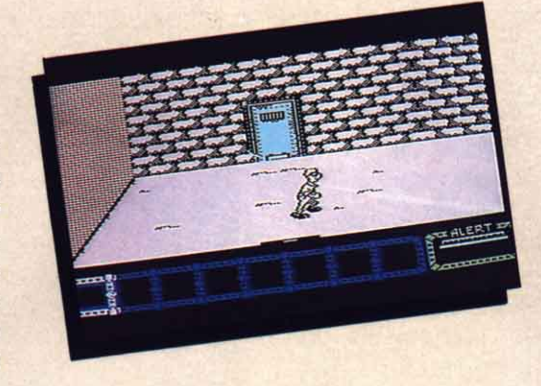


ZOOT es un juego que en un principio despista a quien se atreve a jugar con él. Esto es debido a que Zoot, el protagonista, se desplaza por unas placas móviles, por donde también se desplazan sus enemigos. Estas placas móviles pueden desplazarse tanto de derecha a izquierda como de arriba abajo, pero siempre que Zoot esté sobre la placa a mover y que la desplace hacia un lugar donde no haya otra placa. Estos desplazamientos de placas nos permitirán acceder a lugares donde antes era imposible llegar, o bien cambiar de pantalla o hasta de nivel, si bien antes de abandonar cualquier nivel debemos recuperar las canicas que en él pueda haber. Otro aspecto a tener en cuenta es que a veces nuestros enemigos dejarán de moverse, y entonces, o bien cambiarán de color o bien de forma, de tal manera que no debemos confundirlos con ningún otro elemento del programa, e igualmente destruirlos antes de pasar por donde se encuentran. Los gráficos del programa nos recuerdan a los dibujos de cómics de niños, muy voluminosos y llenos de colores vivos. Quizá por el gran tamaño de éstos el movimiento resulta un poco lento, pero no es excusa para dejar desamparado a Zoot.

ANIMACION	4
INTERES	4
GRAFICOS	5
COLOR	3
SONIDO	2
TOTAL	18

DUSTIN

• DINAMIC ▲ CASSETTE ■ JUEGO





Poco a poco, los programas que DINAMIC va sacando para ordenadores como el Spectrum o el Amstrad aparecen también en una versión para nuestro sistema: el MSX. Y ahora le ha tocado el turno a DUSTIN, una divertida videoaventura, que tiene como protagonista a KID SAGUF. KID SAGUF, más conocido con el nombre de DUSTIN, se encuentra preso en la cárcel de alta seguridad de WADRAS por delitos de robo y falsificación.

Un buen día, harto de estar a la sombra, idea un plan de fuga, que solamente podrá realizar a través de nuestra inestimable ayuda. Lo primero que deberemos hacer es salir de nuestra celda (¡fácil!) y seguidamente conseguir algunos objetos indispensables para fugarnos.

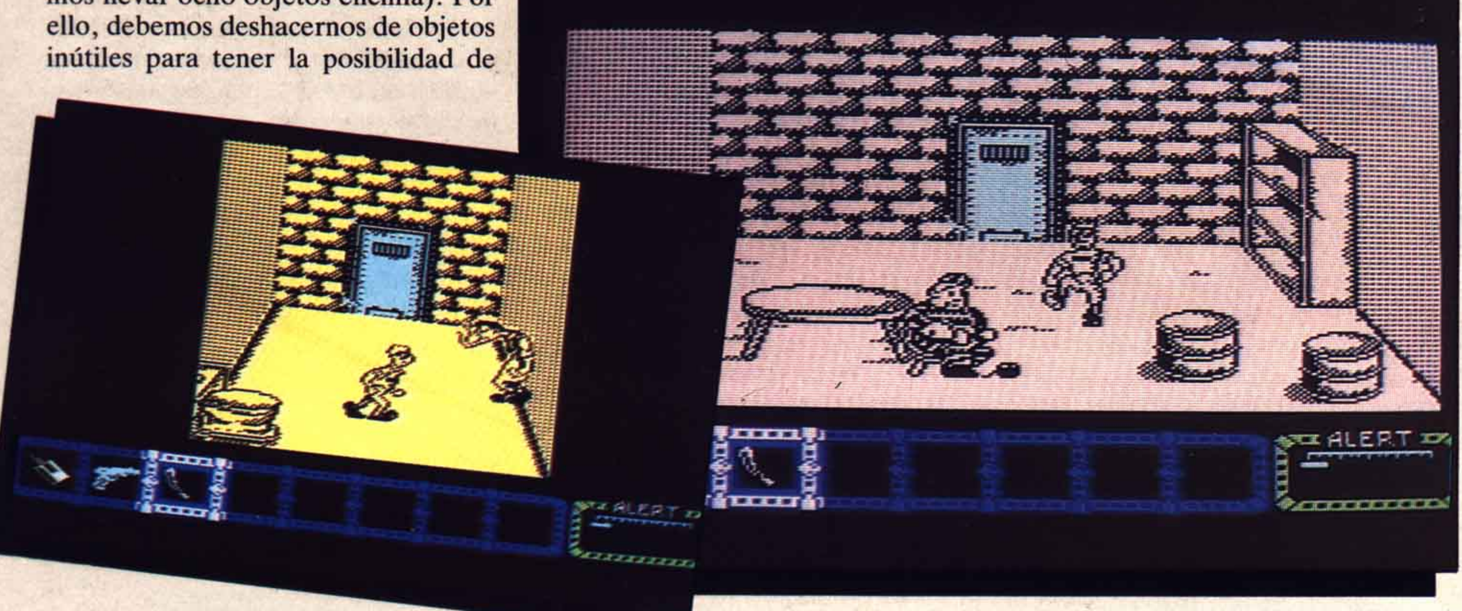
Estos objetos los podemos encontrar en poder de los demás habitantes del centro penitenciario: algunos de ellos nos intercambiarán sus objetos por los nuestros, pero a otros se los tendremos que arrebatar. Sea como sea, una vez que consigamos los codiciados objetos éstos se visualizarán en la parte inferior de la pantalla, en un gráfico rectangular con ocho celdas, cada una de las cuales sólo admite un objeto (es decir, como máximo podemos llevar ocho objetos encima). Por ello, debemos deshacernos de objetos inútiles para tener la posibilidad de guardar otros de mayor utilidad. No obstante, nuestro amigo DUSTIN nos advierte que hay tres objetos que es imprescindible conseguir si queremos lograr fugarnos, y éstos son los siguientes: un hueso, un antídoto y una estatua.

También nos advierte que cada vez que seamos capturados por los guardianes de la prisión éstos nos llevarán de nuevo a nuestra celda y nos requisarán todas las armas que hayamos conseguido. Sin embargo, los objetos obtenidos seguirán en nuestro poder.

Para los que no lo sepáis, os diremos que WAD-RAS es en realidad el nombre de un regimiento donde hacen la «mili» muchos millares de reclutas cada año, entre los que se cuentan, quizá, algunos colaboradores y amigos de la firma Dinamic. De ahí lo de «prisión de alta seguridad de WAD-RAS».

Ésta constituye solamente una prueba más del agudo sentido del humor que caracteriza a estos chicos. Vaya para ellos un saludo y una felicitación cordiales.

ANIMACION	7
INTERES	8
GRAFICOS	8
COLOR	7
SONIDO	6
TOTAL	36

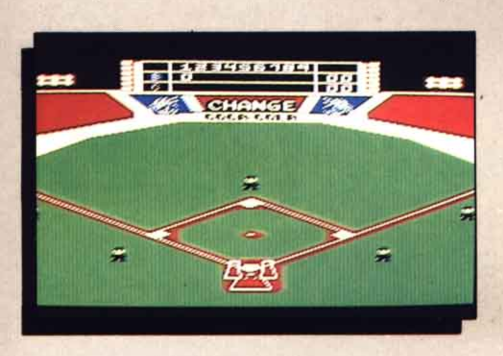


CRAZE BALL

• HUDSON SOFTWARE ▲ TARJETA ROM ■ SIMULACION

Estamos seguros de que éste es el mayor juego basado en el deporte del béisbol que podemos encontrar en el mercado, y no tan sólo en el mercado de los MSX sino también en el de muchos otros ordenadores. Con él podemos realizar todas las jugadas que se efectúan normalmente en un partido de béisbol en vivo, y además toda esta espectacularidad técnica se ve complementada por unas imaginarias estadísticas que podemos cambiar a nuestro gusto.

Una vez cargado el programa, el ordenador nos pide si queremos jugar contra él, jugar la opción de dos jugadores, ver cómo juega él solo contra sí mismo, o bien editar las estadísticas de los jugadores de los dos equipos, el del este y el del oeste, y cambiarlas. Supongamos que escogemos la primera opción: jugar contra el ordenador. Entonces aparecerá en pantalla una panorámica de todo el estadio, saltarán los jugadores a la pista y rápidamente la supuesta cámara a través de la que miramos el partido centrará la imagen en el pitcher. De este modo ahora podemos ver al pitcher rodeado por las cuatro bases, cada una con su correspondiente defensor, y frente al pitcher el bateador del equipo atacante. Además también hay en pantalla la siguiente información: puntuación o carreras hechas por cada equipo hasta ese momento, inning en que nos encontramos, jugador que batea y su estadística, strikes conseguidos por el pitcher, bolas ganadas por el bateador y hombres eliminados del equipo atacante, además de la velocidad con que ha sido lanzada la bola. Todo esto por lo que respecta a la parte técnica del juego.



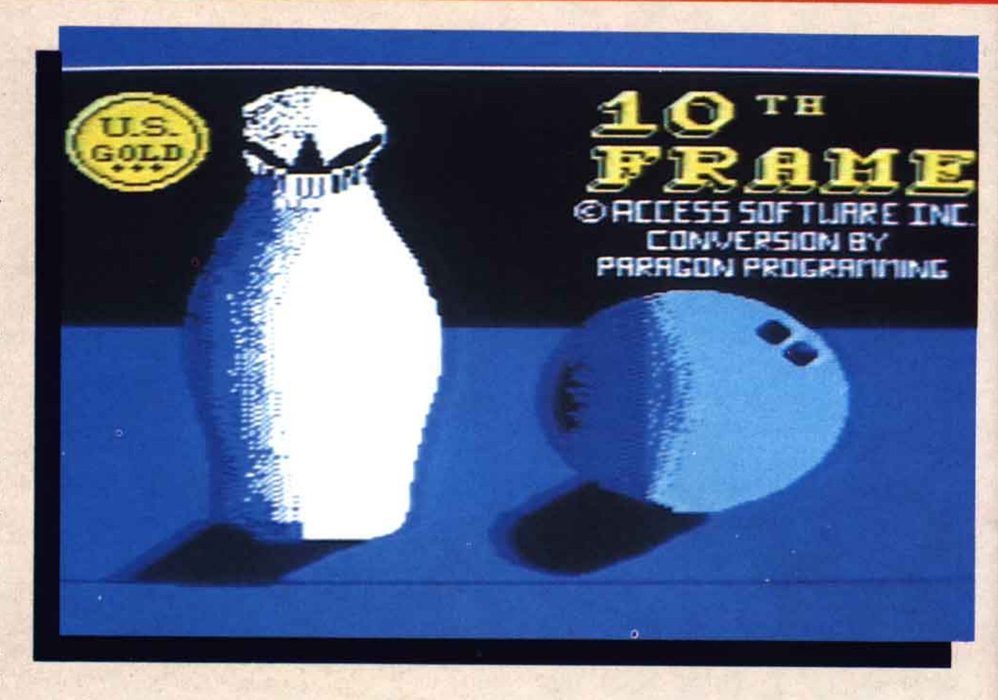
En cuanto a la parte práctica, el programa respeta las tácticas propias de este deporte, y permite realizar todas las jugadas que queramos. Por ejemplo, el bateador puede dar el golpe desde diferentes posiciones de bate, es decir, no es necesario realizar todo el movimiento de brazo para darle a la bola sino que podemos realizar un tiro corto, esperando la bola con el brazo ya extendido. También se puede emplear la vieja táctica de eliminar a los corredores que abandonan la base antes de que el pitcher lance la bola, pasando ésta rápidamente a un defensor de base.

En definitiva, CRAZE BALL es puro béisbol al alcance de todos. Os invitamos a practicarlo.

ANIMACION	7
INTERES	6
GRAFICOS	6
COLOR	6
SONIDO	5
TOTAL	30

10TH FRAME

• ERBE ▲ CASSETTE ■ SIMULADOR



10TH FRAME es un simulador de los juegos de bolos, producido por **ACCES**, casa de Software harto célebre por los BEACH HEAD, LEADER BOARD, etc.

Antes de referirnos a esta versión informatizada de este popular juego, te ilustraremos –o recordaremos– su historia y fundamentos.

Los bolos es el deporte de cancha cerrada que más aficionados cuenta en Estados Unidos. Fue introducido por los holandeses en la isla de Manhattan (Nueva York) en 1623. Sus normas se dictaron en 1895.

En este deporte el jugador lanza una bola hacia un grupo de diez bolos dispuestos en triángulo, con objeto de derribar el mayor número posible de ellos.

Los bolos se disponen al final de un pasillo. A ambos lados de la pista existen unos canales para recoger las bolas que se desvían. Se juega con bolas de madera de guayaco, con dos o tres agujeros para introducir los dedos. El *strike*, la mejor jugada, consiste en derribar todos los bolos al primer golpe. El score máximo es de 300 tantos. Gana la partida el jugador que haya conseguido mayor tanteo.

10TH FRAME ofrece la posibilidad de participar en un OPEN DOMESTICO DE BOWLING, incluyendo hasta 8 jugadores y hacer liguillas creando equipos entre los jugadores.

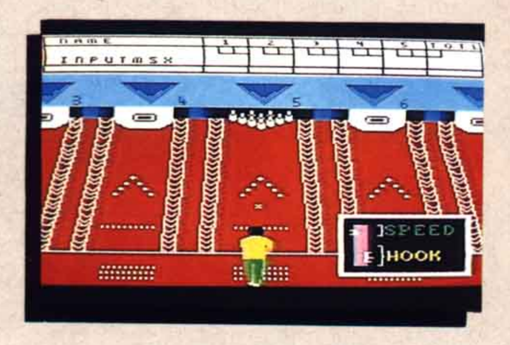
Hay tres niveles de dificultad que van desde el **KIDS** (niños) hasta profesional, pasando por amateur. El juego es muy sencillo: cada jugador tiene ante sí una marca que le ayuda a seleccionar el ángulo de tiro. El otro parámetro es la posición que cada jugador tiene con respecto a la pista.

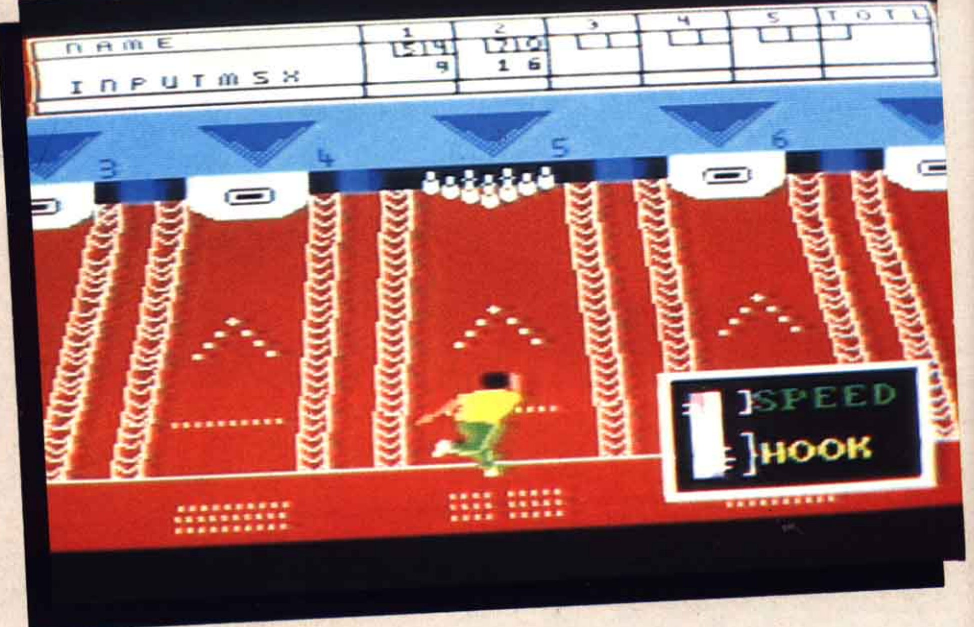
El juego se complica enormemente cuando aparecen los efectos (llamados aquí «tiro de gancho») y el modo de obtenerlo, que es harto difícil.

Sin embargo, a pesar de los evidentes atractivos que 10TH FRAME pueda tener para los que seáis muy forofos del juego de los bolos, no hemos podido menos que sentirnos levemente decepcionados ante este simulador de la casa ERBE.

En efecto, solamente podemos disponer de una pantalla de juego, y la animación, además, no es la que nos tiene acostumbrados la casa.

ANIMACION	8
INTERES	6
GRAFICOS	7
COLOR	7
SONIDO	6
TOTAL	**34**





ZOIDS! THE BATTLE BEGINS

MARTECH ▲ CASSETTE ■ ARCADE

Zoids es un juego adictivo y con un gran argumento digno de **Asimov**.

Zoids Star es una estrella habitada por organismos vivientes que temen por su seguridad y dedican largos años a la investigación de técnicas avanzadas de guerra.

Cuando acaban con el proceso de desarrollo, deciden crear androides para las fortalezas volantes que han diseñado. Androides bautizados como **Zoids** y que son destinados a dos tareas: luchar entre sí como gladiadores y patrullar por los alrededores del planeta, previniendo posibles invasiones sorpresa.

Lamentablemente, un meteorito de dimensiones gigantes viene a perturbar esta paz, haciendo desaparecer a los organismos vivientes y dejando el planeta en manos de los androides.

Antes de morir, uno de los organismos vivientes lanza un SOS, que es recogido por una de las naves patrulleras. En la excitación del momento, un mal cálculo de trayectoria hace que esta nave de salvamento se estrelle en la Luna azul, único satélite de **Zoids Star**. La diferencia de temperaturas y radiaciones de la Luna azul hace que los supervivientes del accidente tengan que volver a diseñarse ellos mismos, adquiriendo un color rojo. Del día a la noche estos **Zoids** rojos, capitaneados por **Rojo el Terrible**, se vuelven malos, muy muy malos.



Deciden unificar fuerzas, conquistar **Zoids Star** convirtiendo a los otros androides (de color azul) en esclavos.

Ante la amenaza, los azules se agrupan y construyen una nave insignia llamada **Zoidszilla**, que hace que las fuerzas sean más equitativas entre ambos bandos. En ese momento una pequeña nave espacial se estrella en **Zoids** y el único superviviente humano se inserta en la sociedad de los **Zoids** azules y les ayuda a trazar un plan: actuar como espía, entrar en la ciudad de los **Zoids** rojos y dejar allí un veneno que acabe con tan pérfidos y ominosos entes; lamentablemente un misil lanzado por el propio **Rojo el Terrible** acaba con la nave insignia **Zoidszilla** y sus tripulantes (el hombre incluido). Las seis piezas principales de la nave son recogidas por **Rojo el Terrible** y puestas en las seis puertas de su ciudad.

El ánimo de los **Zoids** azules está en controlar sólo la inesperada aparición de otro hombre y su empeño por recuperar las seis piezas de la nave, las únicas que devolverán a los **Zoids** azules el valor y el potencial de guerra que pueden salvar al planeta de la colonización. Este hombre eres tú.





Con esta trama que parece salida de la pluma de Isaac Asimov o de Ray Bradbury, y tú convertido en un auténtico Harrison Ford enfrentado a estos siniestros "replicantes" –como en la notable película de ciencia ficción **Blade runner**–, tendrás la ocasión de demostrar ampliamente tu entereza, valor y gallardía ante la pantalla de tu ordenador MSX.

ANIMACION	6
INTERES	5
GRAFICOS	5
COLOR	6
SONIDO	6
TOTAL	**28**

FLASH GORDON

• MASTERTRONIC ▲ CASSETTE ■ ARCADE



FLASH GORDON, el famoso y rubio héroe intergaláctico, luego de triunfar y vivir infinidad de fantásticas aventuras en las pantallas de cine y en los cómics, llega a la VDU de tu ordenador gracias a este arcade de la casa Mastertronic. El juego consta de tres partes muy diferenciadas. En la primera parte debes penetrar por una jungla y orientarte hacia el norte. Tienes 24 horas para llegar a la fase siguiente, en el camino verás cómo esta jungla está plagada de **OURANG OUTANES**, arañas peligrosas, pajarracos difíciles de identificar y trampas en el suelo que deberás evitar dando unos saltos dignos de BRUCE LEE.

Como única arma dispones de un par de COLTS cuyos barriletes de balas aparecen en la parte inferior derecha de la pantalla. Un golpecito de *joystick* hacia arriba y verás cómo salen las balas del barrilete.

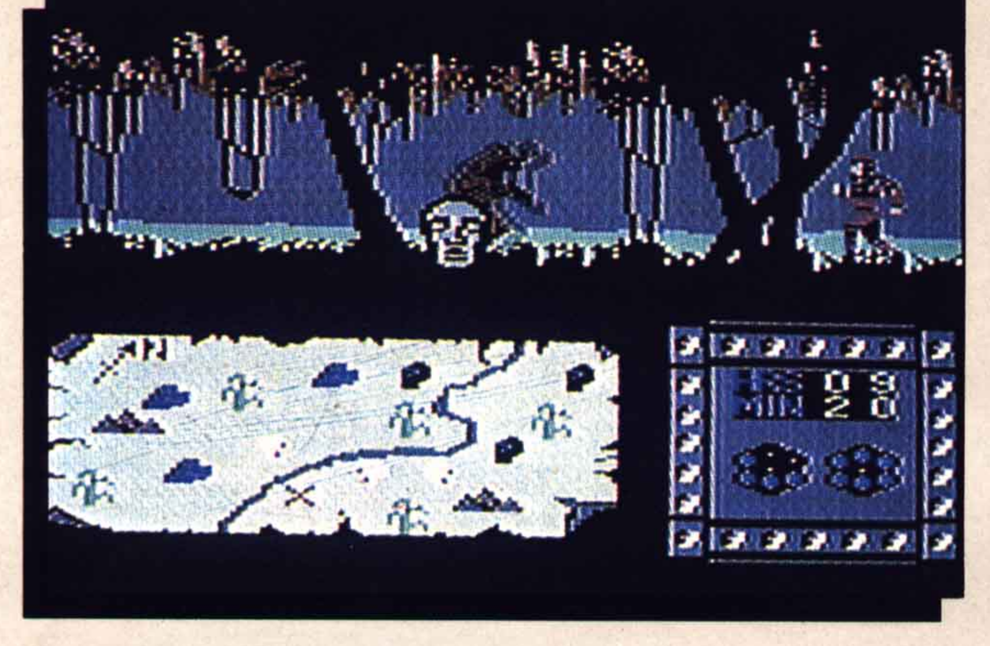
En esta fase será de vital importancia que realices un mapa del juego ya que sin él no pasarás de fase ni de broma. En tu recorrido verás cómo aparecen cajas en el suelo; acércate a ellas, pues te repondrán fuerza o bien te darán más balas para disparar.

La segunda fase te enfrenta al malvado BARING en una lucha a muerte en las mazmorras del planeta. En esta fase el juego se convierte en un auténtico EXPLODING con una serie de golpes permitidos: patada a la cara, voltereta con golpe al rostro, etc...

En la parte inferior aparecerán los dos oponentes tirando de una cuerda que indica el nivel de fuerza de cada uno.

Si logras permanecer vivo en esta ocasión, verás cómo pasas a la siguiente opción: la moto. Deberás conducir una moto a la máxima velocidad a derecha e izquierda y el vehículo se ladea; verás de vez en cuando aparecer unos *flashes* en la pantalla: eso es indicador de que **el malvado Ming** está delante de ti en el ángulo de los disparos... Si logras hacer todo y vencer en esta última fase estará muy bien, ya que significará que has salvado el mundo.

Claro que si eres un jugador empedernido estarás ya muy acostumbrado a ello.



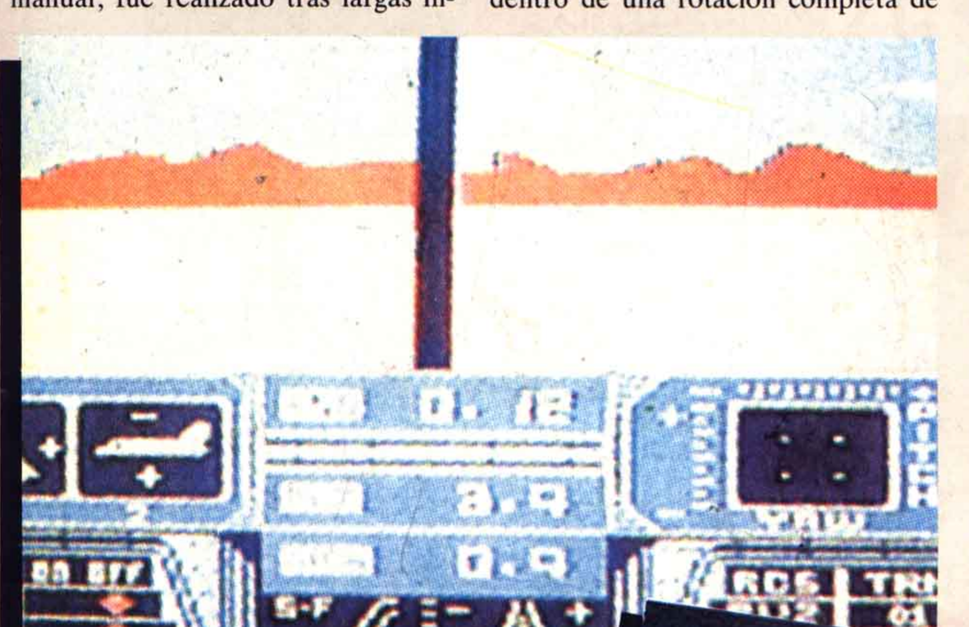
ANIMACION	7
INTERES	7
GRAFICOS	5
COLOR	6
SONIDO	5
TOTAL	**30**

SPACE SHUTTLE

• ACTIVISION ▲ CASSETTE ■ SIMULADOR

SPACE SHUTTLE es un simulador de vuelo de la nave Columbia; el objetivo es sencillo: tratar de llegar lo más cerca posible de la órbita de un satélite geoestacionario.

SPACE SHUTTLE, según reza el manual, fue realizado tras largas investigaciones en la NASA para que el juego tuviera el mayor grado de realismo.



La verdad es que este simulador consigue su propósito plenamente.

Dispones de varias opciones de vuelo que corresponden a varias opciones de destreza (comandante, piloto, especialista de misión, y especialista en instrumentos).

Asimismo, los diferentes niveles de dificultad coinciden con condiciones de vuelo muy diferentes, y con actuaciones (correcciones) de la computadora de a bordo muy distintas. En el nivel más sencillo de vuelo no se tendrán en cuenta los innumerables factores adversos que pueden, en la realidad, hacer que se retrase o posponga un vuelo: ni importará demasiado que haya un viento huracanado, que en la pista la visibilidad sea únicamente de 3 o 4 metros, o bien que unas borrascas hayan llenado la pista de agua...

Estos detalles te serán comunicados por la pantalla de a bordo.

La pantalla recrea la vista del piloto de la nave y en ella se distinguen varios monitores de vuelo: las tres fases del lanzamiento de la lanzadera, con una recreación en pantalla de la situación de la nave por medio de radar. Otro indicador te dirá cuál es tu posición exacta con respecto a la del satélite, dentro de una rotación completa de nuestro planeta y, por último, podrás ver en miniatura una pantalla de ayuda que te será muy útil si quieres volver a pisar tierra firme.

Junto con estos instrumentos podrás ver indicadores de ejes X, Y.

Así como del estado de la nave, dispondrás de una panoplia de chivatos de avería muy sofisticados. Entre ellos destacan por su extraordinaria eficiencia los siguientes: Sistema OMS: sistema de maniobras orbitales usado por la auténtica nave y que te brindará la oportunidad de realizar una mejor aproximación al satélite, sistema de ayuda por medio de un auténtico piloto automático digital, etc...

En los niveles más difíciles de juego, como el de comandante, verás que tendrás que tomar una serie de decisiones.

En este nivel de juego el ordenador no te dejará pasar ni un solo error: en cualquier decisión el ordenador asignará una probabilidad de éxito o fracaso y te dejará o no seguir según lo asignado. Así que si adoptas la determinación de despegar mientras haya borrasca, puede que el ordenador te deje hacer los 500 primeros metros pero luego, como por arte de magia, deberás corregir en un momento un cambio de trayectoria...

Los gráficos son buenos y la velocidad de *scroll* dota a este simulador de un gran realismo.

Como verás, este simulador te permitirá convertirte, si eres perseverante, en un auténtico émulo de Neil Armstrong.

ANIMACION	7
INTERES	8
GRAFICOS	7
COLOR	7
SONIDO	6
TOTAL	**35**

Dibuja la MaSXcota de INPUT





Más de un año en onda programera y nosotros sin MaSXcota.

Un nuevo y delirante concurso sale a la luz para dotar de premios sustanciosos a aquellos programadores en BASIC y Código Máquina (de primera y segunda generación) que gustan de «pintar monigotes» en la pantalla.

Las bases son sencillas: envía tus dibujos hechos por ordenador de la que piensas podría ser nuestra/vuestra mascota (cuantos más, mejor) en cinta o diskette 3 1/2 a:

La Mascota de INPUT
Aribau, 185, 1.º
08021 Barcelona

Entre los seleccionados de cada mes se sortearán tres fantásticos programas en cartucho, entrando a formar parte como finalistas en el sorteo que tendrá lugar a fin de año.



EL ZOCO

Intercambiamos programas en disco 3 1/2 para MSX-1 y MSX-2 tales como: CABBAGE KID, J.P. WINKLE, NINJA PRINCESS, STAR FORCE, TEARS OF NILE, WHO WHO WARS, YOUKAI, BOKOSUKA WAR, MR DO VS UNICORN, MOON SWEEPER, SKY GALDO, STAR SOLDIER, ELEVATOR ACTION, HANG ON, GUNDAM, SCARLET 7. Buscamos desensambladores, proc. textos, utilidades para disco y cartucho. No serios abstenerse. INPUT SOFT PROGRAMMERS. Doctor Ferrán 21. HOSPITALET LLOB. (Barcelona).

Cambio juegos MSX, como Arkanoid, Zanac, etc... José Herminio Gascón Ortega. C/ G. Vía Germanías n.º 33, 8.º pta. n.º 16. Tavernes de Valldigna. Valencia. Tel: (96) 282-16-37

Cambio juegos. Santi Sandoval. C/ Rambla Justo Oliveras n.º 69. Hospitalet. Barcelona. Tel: (93) 337-36-37

Vendo ampliación de memoria MSX 64 K.. Ismael Tel: (952) 49-02-42 de 17:00 h. a 22:00 h.

Intercambio juegos para MSX, tengo Batman, G.P., Dam Busters, etc., y me gustaría tener Livingstone, Spirits, Army Moves, etc. Javier Payarols Caro. C/ St. Fructuoso n.º 51, 5.º Tel: (93) 424-32-22 08004 Barcelona.

Intercambio información y programas MSX, en disco 3,5" y cinta. Luis Alberto Pérez Pérez. C/ Ebro n.º 7, 5.º B 47013 Valladolid. Tel: (983) 23-59-36

Vendo cintas originales de los juegos: Phantomas II, Bounder, Chiller, Storm, Sorcery, más 30 copias de buenos juegos. José. Tel: (971) 55-33-64

Compro ampliación de 64 K. Preguntar por Luis a partir de las 19:00 h. Tel: (93) 385-93-70

Vendo ordenador SONY MSX P-20, en buenas condiciones, lleva dos meses de uso y con una garantía total de seis meses; también incluyo cassette y más de 10 juegos. José Antonio Martínez González. C/ Crrt de Beniel n.º 76. Alquerías. Murcia. Tel: (986) 81-25-18

Cambio juegos MSX, poseo bastantes. Richard González Parada. Can Bordoi. 08450 Llinars del Vallés. Barcelona. Tel: (93) 841-08-75

Contacto con usuarios de unidades de disco tanto MSX-1 como de MSX-2, con el fin de intercambiar experiencias. M.E. Martínez. C/ Alfonso I, n.º 28, 6, B. 50006 Zaragoza

---

Cambio juegos de primeras marcas, Replicante y Diskette. Joan Miranda. C/ Monflorit n.º 59 Gavá 08850 Barcelona. Tel: (93) 662-15-49

Intercambio juegos, poseo últimas novedades. Oscar Tel: (986) 88-29-82

Cambio juegos en disco. Francisco Servera. C/ Serrería n.º 18. Ploblença Mallorca. Tel: (971) 53-28-72

Intercambio juegos MSX, tengo últimas novedades; sólo para Salamanca Juan José García Cobreros. C/ Av. Mirat n.º 3-7. Tel: (923) 25-03-06

Intercambiamos juegos MSX con garantía total. Jordi Agramunt Soler. C/ Av. Meridiana n.º 488. 08030 Barcelona. Teléfono (93) 346-01-30 de 20.00 h. a 23.00 h.

Intercambio programas MSX, tengo Arkanoid, Zanac, etc., y me interesan Samantha Fox, Raid on Bungeling, entre otros. José Davis Castellanos Vela. C/ Horno n.º 19. 13600 Alcázar de San Juan. Ciudad Real. Tel: (926) 54-42-97

Desearía crear un club de MSX en Socuéllamos para intercambiar pokes, programas e ideas para el ordenador. Davis Salmerón Campos. C/ Generalísimo n.º 2. Socuéllamos Ciudad Real. Tel: (926) 53-02-75

Cambiamos juegos MSX. Deseamos entre otros Penguin Adventure, Yie Are Kung-Fu II, etc., también queremos formar un club a nivel nacional. Manuel Angel Pajuelo. C/ Teatro n.º 15. Pya. Pueblo Nuevo. 14200 Córdoba. Tel: (957) 56-06-47

Vendo Spectravídeo 728 MSX 80 K, teclado profesional, cables, grabadora especial ordenador, manuales, libros, más de 50 revistas, más de 100 programas y buenos juegos, todo por sólo 47.000 ptas. Vendo opcional TV b/n por sólo 7.000 ptas. Juan Pablo Singer. C/ Av. República Argentina n.º 37 2.º 2.ª. 08023 Barcelona. Tel: (93) 211-16-31

Cambio Juegos MSX, me interesan Green Beret, Soccer, etc., además vendo cartucho Roger Rusbish en perfecto estado o cambiarlo por el Némesis, además compraría unidad de disco de 3'5 pulgadas. Eduardo. Tel: (976) 65-22-23 de 13.00 h a 15.00 h.

Deseo intercambiar juegos MSX. José Manuel Martín Ortega. C/ Bda. La Paz Bque. n.º 8 4.º F C/ Stravinsky 29004 Málaga, Tel: (952) 33-21-10

Intercambio juegos, poseo las últimas novedades. Alfonso. Urb. Miraflores, Torre n.º 7 Pl. 2.ª Pta. 3.ª Marbella 29600 Málaga. Tel: (952) 820982

Intercambio programas MSX o MSX-2 en disco. Francisco Daza Guirao. C/ Calderón de la Barca n.º 101 At. 2.º Esc. Izq. Barcelona. Tel: (93) 397-48-26

Vendo Spectravídeo 728 MSX de 80 K, con cables, libros, unidad de Diskettes Sony MSX HB D-50 de 3'5 pulgadas, con un mes de uso. Lo vendo junto o separado y todo muy barato, por la compra regalo más de 200 juegos. Xavier Martínez Vidal. C/ Emérita Augusta n.º 10 esc. B 3.º Barcelona 08028. Tel: (93) 339-86-22

Cambio programas de actualidad. Luis Manuel Pérez Ramos. C/ Barberito Edif. Sta. María 7.º 2.ª 29003 Málaga

---

Cambio juegos MSX, mandar lista a Jordi Díaz C/ Leiva n.º 18 4.º 1.ª. 08014 Barcelona Tel: (93) 431-71-36

Cambio programas MSX-1 a escoger entre más de 350, pedir lista a Jordi Arias Ruiz. Tel: (93) 654-48-22. C/ Abat Escarre n.º 20-22 3.º 1.ª. 08830 Sant Boi. Barcelona

EL ZOCO

Cambio programas de todo tipo, poseo más de 200, con las últimas novedades. Guillermo Pérez Lojo. Apdo. 33. 15961 Riviera. La Coruña. Tel: (981) 87-10-78

Cambio juegos primeras marcas. También puedo cambiar los juegos por un cartucho de ampliación de 16K o 32K. Xavier Gascón Vega. C/Rosario Sta. Fe n.° 2. 08338 Premià de Dalt. Barcelona Tel: (93) 751-00-28

Cambio juegos MSX, poseo unos 70 como Arkanoid, King's Mare. Angel Vicente Herguedas. C/ Pintor Velázquez n.° 38, Pta. 12. 46100 Burjasot. Valencia. Tel: (96) 374-37-18

Intercambio toda clase de juegos para la primera y segunda generación de MSX, en disco de 3,5 pulgadas. Llamar a Carles Baltrons. Teléfono (972) 33-13-13.

Cambio los siguientes programas Krakout, Fides Monty, Last Mission, Army Moves y Nonamed. Antonio Montero C/ Ciudad Cooperativa n.° 49 5.° 2.ª. 08830 Sant Boi. Barcelona. Tel: (93) 652-16-94

Cambio programas MSX de todo tipo, como Bounder, The Dam Busters, etc. Miró Tel: (981) 82-71-05

Vendo ordenador sony HB-75P prácticamente nuevo, regalo 100 juegos con él. También cambio juegos. Agustín López Neira. C/ Alta n.° 50 3.° H 39008 Santander. Tel: (942) 37-21-86

Cambio juegos, poseo de última novedad. David Landaluce Aragón C/ Pl. de La Concordia n.° 6 Dcha. 6.°A Marbella. Málaga.

Vendo ordenador PHILIPS VG-8020, cables de conexión, joystick, libro «MSX BASIC» y 20 juegos que tú eliges, todo por 40.000, en buena conservación. Sebastián Moya Carballo. C/ Camino de los Reyes s/n. La Cuesta Tenerife

Cambio programas MSX, busco Perry Mason, Las Vegas, etc., poseo entre otros Gauntlet, Soccer, etc. Francisco Pérez Galea. C/ Concepción Arenal n.° 50 1.° B 06480 Montijo. Badajoz. Tel: (924) 45-15-26

Vendo Ploter Sony Color PRN-C41, completo con alimentador, cables de conexión instrucciones en español y un cartucho de dibujo, precio a convenir. Antonio Tel: (93) 674-87-11 a partir de las 20.00 h.

Cambio programas MSX en disco de 3'5", tengo 200. Fco. López. C/ La Carrera n.° 34. 12530 Burriana. Castellón. Tel: (964) 51-18-75

Desearía contactar con usuarios de MSX con el fin de intercambiar ideas, pokes, programas y potenciar nuestro sistema en Logroño. Sergio Ascaso C/ General Franco n.° 48 3.° Dcha. Tel: (941) 24-51-84. Logroño (La Rioja)

Intercambio programas MSX y MSX-2 preferiblemente en disco de 3'5" Mandar lista indicando los que se posean en disco, garantizo contestación. Daniel Cortés Zaragoza. C/ Escultor Peresejo n.° 26 7.° 03800 Alcoy. (Alicante)

Cambio programas MSX, tengo más de 300. Pedro Bofarull. Tel: (977) 31-32-32

Vendo ordenador Sony HB-75 de 80K, manuales, cables, joystick y 100 juegos, todo por 45.000 ptas. Angel Vicente Herguedas. C/ Pintor Velázquez n.° 38 Pta. 12 46100 Burjasot. Valencia. Tel: (96) 364-37-18

Intercambio juegos MSX, tengo Army Moves, Nonamed y muchos más Oscar Guillén Pay. Apartado 123. Archena (Murcia)

Cambio juegos y utilidades con usuarios de MSX, dispongo de muy buenos programas. Gabriel Bustos Martín. C/ Circunvalación n.° 28. 23300 Villacarrillo. Jaén. Tel: (953) 44-00-60, preferiblemente por las mañanas

Intercambio programas MSX-1 en disco, poseo más de 450 títulos. Oscar López Pérez. C/ Alberche n.° 136 Edif. Granada 1-A. 45007. Toledo. Tel: (925) 23-26-84

Intercambio juegos y utilidades para MSX-1 y MSX-2, en disco de 3'5", poseo más de 300. Juan M.ª Gorrotxateguir. C/ Guipúzcoa n.° 16 4.°C. 20240 Ordizia. Guipúzcoa. Tel: (943) 88-54-74

Intercambio toda clase de juegos, poseo Dunk Shot, Knight Mare, etc. Daniel Jiménez Mouriño. C/ Travesía de Vigo n.° 72 6 D. Vigo. Pontevedra. Tel: (986) 27-84-49

Vendo ordenador Sony HB-55P con Data Cartridge, cartucho ampliación, cables de conexión, manuales en castellano y 20 juegos por 23.000 ptas. Antonio Fortea Mingarro. C/ Camí D'Onda n.° 36 7.°. 12530 Burriana. (Castellón)

Intercambio programas para MSX con toda España, poseo más de 100 juegos de las primeras marcas. Mariano Galvache Murillo-Rico. C/ La luna n.° 108 Pya. Pueblo Nuevo. Córdoba. Tel: (957) 56-21-12

Deseo conseguir ordenador y periféricos. Armando. Apartado 274. 45600 Talavera (Toledo)

Vendo ordenador PHILIPS MSX VG-8020 de 80 K RAM, manuales, cables y 200 programas, así como unidad de disco Sony HBD-50, precios a convenir. Jorge Esteve Figueras. C/Más Casanovas n.° 8-10 3.° 3.ª. 08025 Barcelona. Tel: (93) 235-64-88

Cambio juegos Zanac, Arkanoid, etc. Javier Vizoso Guerra. C/ Travesera de la Obra n.° 1 1.° Iz. Sada. (La Coruña). Tel: (981) 62-04-43 de 13.30 a 15.30

Cambio programas, últimas novedades, poseo más de 500 programas José L. Rus. C/ Colón n.° 27 2.°A. 08400 Granollers. Barcelona.

Cambio programas de todas clases MSX, tengo entre otros Soccer, Green Beret, etc. Luis Javier Navarro Marín. C/ Regente Molina Valero n.° 3 23300 Villacarrillo. Jaén. Tel: (953) 44-00-61

DYNADATA

A LA VANGUARDIA DE MSX

APRENDA INFORMATICA EN CASA

Con el nuevo programa autodidáctico audiovisual de DYNADATA, usted puede aprender cómodamente en su casa los fundamentos de la informática y también a programar en el lenguaje común, conocido como BASIC. El curso completo consta de 12 cassettes audiovisuales, teniendo una presentación en pantalla muy completa y amena, permitiendo seguir las instrucciones en forma sencilla, parando y poniendo en marcha el sistema según el deseo y progreso del interesado.

El DYNADATA DPC-200 es como aprender en un ordenador personal IBM debido a la similar posición de letras y signos.

LECTORA DE DISCOS DE 1.5 MBYTES

DYNADATA, en su continuo afán de desarrollo en el campo del MSX, ha presentado una nueva lectora de discos con capacidad de 1.5 Mbytes. Esta lectora se incorpora al ordenador MSX, permitiéndole la lectura de programas bajo los sistemas operativos MSX-DOS, CP/M y MSX-BASIC. Tiene en su configuración dos discos de 3,5 pulgadas, con una nueva capacidad de 750 Kbytes cada uno.

DYNADATA, con esta nueva lectora, sigue brindando al ordenador MSX mayor potencia operativa a precios muy económicos y al alcance de cualquier usuario.

AULA INFORMATICA

El aula informática DYNADATA es un sistema didáctico interactivo que cuenta ya con varias instalaciones en toda España, con plena aceptación por parte de los colegios e institutos de segunda enseñanza.

Es un sistema que controla desde una unidad central, atendida por el profesor, 32 puestos autónomos. En su mesa, éste cuenta con controlador, tres unidades de disco, ordenador DYNADATA MSX, DPC-200, con teclado en español; una impresora Centronics de 180 cps. y otra lectora de disco de 360 Kbytes, para que cargue los programas desde su puesto a las unidades del controlador central, supervisando así todos los puestos de los alumnos, admitiendo comandos especiales para la transferencia de pantallas, programas en lenguaje BASIC y código de máquina y de información interna de los puestos, de una manera interactiva, a través de mensajes recíprocos con los alumnos.

Los pupitres de los alumnos comprenden dos puestos de trabajo y contienen dos ordenadores DYNADATA MSX, DPC-200, con sus respectivos monitores de fósforo verde o color, y dos unidades de cassette modelo C-683B, que permiten el almacenamiento de programas para poderlos utilizar fuera del aula escolar.

Hay que destacar que el ordenador DYNADATA MSX es el único en el mercado con la posición de las teclas idénticas al IBM PC, logrando con esto que un cambio posterior a un ordenador profesional no suponga esfuerzo alguno.

Para más información llamar a los teléfonos:
(91) 279 21 85 - 279 28 01 - 270 50 07

OFERTAS MSX

ORDENADOR + CASSETTE + 2 JOYSTICK .... 33.900 Pts.
ORDENADOR + CASSETTE + 2 JOYSTICK + MONITOR FOSFORO VERDE ... 46.600 Pts.
ORDENADOR + MONITOR FOSFORO VERDE + LECTORA DE DISCOS .... 83.000 Pts.
CURSO AUTODIDACTICO (12 CASSETTES, AUDIOVISUAL) 14.400 PTAS.
CON LA COMPRA DE ALGUNA DE LAS OFERTAS ANTERIORES
(SU PRECIO HABITUAL ES DE 28.000 Pts.)
IMPRESORA + PROCESADOR DE TEXTO .... 14.400 Pts.
UNIDAD DE DISCO (PARA DYNADATA O SPECTRAVIDEO) .... 44.700 Pts.

NOTA: Lectora de discos de 750 Kbytes, 3,5 pulgadas.

NUEVO SOFTWARE COMPILADOR DE MSX BASIC

Un compilador que traduce programas escritos en el lenguaje de alto nivel BASIC a códigos, directamente ejecutable por el microprocesador, lenguaje máquina. La ejecución es más rápida y el consumo de memoria es menor.

MSX-PLAN

Una hoja de cálculo completa para su ordenador MSX de primera generación. Tiene un amplio juego de operaciones sobre las celdillas y una colección de operadores matemáticos sumamente extensa. Guarda bastante similitud de operación con las hojas de cálculo de ordenadores profesionales.

Permite el intercambio de datos con otras hojas de cálculo gracias a un módulo de unión simbólica. Este también funciona como puente con sus programas en MSX-BASIC.

MSX-WRITE

Un sencillo pero completo procesador de textos que le permitirá confeccionar de una manera cómoda y rápida sus documentos.

Como unidad de almacenamiento, admite tanto el cassette como la unidad de disco, lo que le permite ajustarse a todo tipo de economías.

Funciona con pantalla de 40 columnas para poder trabajar con TV. o monitor.

Tiene un sistema propio de gestión de impresora para permitirle cuidar la impresión al mínimo detalle.

NOVEDADES

TARJETA DYNADATA DE 80 COLUMNAS PARA MSX Y CP/M

Los ordenadores de primera generación de MSX podrán utilizar la biblioteca de gestión de CP/M, antes inaccesibles, pudiendo aprovechar también software de segunda generación de MSX, que, de otra manera, no está a su alcance.

El Departamento de Hardware de DYNADATA ha desarrollado la nueva tarjeta de 80 columnas, que permite a un ordenador doméstico MSX de primera generación utilizar programas de 80 columnas bajo los sistemas operativos MSX-DOS y CP/M.

MACROENSAMBLADOR PARA MSX

Ahora existe MSX-MACRO, potente herramienta de desarrollo de software para MSX. Incorpora todas las facilidades de otros ensambladores de muy alto nivel. Destaca como su principal característica la posibilidad de ensamblado interactivo, que permite la modificación de un programa fuente durante el ensamblado. Dentro de las aplicaciones de este ensamblador está un pequeño compilador de Tiny BASIC, posible gracias a la estructuración de la programación en lenguaje máquina. Su incorporación en el paquete, junto con su código fuente, le será posible desarrollar un compilador para su versión particular y personalizada del lenguaje BASIC.

...¡CAMBIA DE ONDA!..



SONY

Konami SHOP

te dan hasta 13.000 pts. por tu viejo ordenador o consola al comprar un ORDENADOR MSX 1 o 2

Konami shop. Francisco Navacerrada 19. 28028 Madrid. Telf. 2557563